

# THÈSE

défendue par  
Étienne Simon

en vue de l'obtention du grade de Docteur

---

## DEEP LEARNING FOR UNSUPERVISED RELATION EXTRACTION

---

soutenue publiquement le 5 juillet 2022

Devant le jury composé de

**Pr Alexandre Allauzen**

Professeur des universités, Université Paris-Dauphine PSL, ESPCI

**Dr Benoit Favre**

Maître de conférences, Aix-Marseille Université

**Pr Pascale Sébillot**

Professeure des universités, IRISA, INSA Rennes

**Pr Xavier Tannier**

Professeur des universités, Sorbonne Université

**Dr Benjamin Piwowarski**

Chargé de recherche, CNRS, Sorbonne Université

**Dr Vincent Guigue**

Maître de conférences, Sorbonne Université

Rapporteur

Rapporteur

Examinatrice

Président

Directeur

Directeur



# Abstract

Capturing concepts' interrelations is a fundamental of natural language understanding. It constitutes a bridge between two historically separate approaches of artificial intelligence: the use of symbolic and distributed representations. However, tackling this problem without human supervision poses several issues, and unsupervised models have difficulties echoing the expressive breakthroughs of supervised ones. This thesis addresses two supervision gaps we identified: the problem of regularization of sentence-level discriminative models and the problem of leveraging relational information from dataset-level structures.

The first gap arises following the increased use of discriminative approaches, such as deep neural network classifiers, in the supervised setting. These models tend to collapse without supervision. To overcome this limitation, we introduce two relation distribution losses to constrain the relation classifier into a trainable state. The second gap arises from the development of dataset-level (aggregate) approaches. We show that unsupervised models can leverage a large amount of additional information from the structure of the dataset, even more so than supervised models. We close this gap by adapting existing unsupervised methods to capture topological information using graph convolutional networks. Furthermore, we show that we can exploit the mutual information between topological (dataset-level) and linguistic (sentence-level) information to design a new training paradigm for unsupervised relation extraction.

0055  
0056  
0057  
0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
0080  
0081  
0082  
0083  
0084  
0085  
0086  
0087  
0088  
0089  
0090  
0091  
0092  
0093  
0094  
0095  
0096  
0097  
0098  
0099  
0100  
0101  
0102  
0103  
0104  
0105  
0106  
0107  
0108

0109  
 0110  
 0111  
 0112  
 0113  
 0114  
 0115  
 0116  
 0117  
 0118  
 0119  
 0120

## Acknowledgements

0121 I'm not sure how to write these acknowledgements who to thank how to thank them there are so many people  
 0122 who contributed to the completion of this thesis maybe I should thank my optician who enabled me to read  
 0123 so many papers but why stop there I'm sure my glasses were made using some kind of polishing machine with  
 0124 bolts made by a worker supported through their childhood by a sweetheart now long forgotten I'm however  
 0125 deeply thankful to the boltmaker's childhood sweetheart for bringing them a bit of warmth that might not  
 0126 have been strictly necessary to the conception of the metal fasteners indirectly ensuring my optical prowesses  
 0127 but that I honor anyway it might be easier to list people I don't want to thank no that would be ill-disposed  
 0128 I won't go into woollen coats I'll focus on people more closely related to my doctoral endeavors first of which  
 0129 are my supervisors thank you Benjamin Piwowarski for providing key insight on the information extraction  
 0130 field the gentle encouragements and for the guidance through the years thank you Vincent Guigue for your  
 0131 vast knowledge of how things are done and how they are not I hope you do not disapprove too much of  
 0132 the otters in my defense's slides I'm also grateful to other members of the jury for examining my thesis in  
 0133 particular I would like to thank the reviewers for thoroughly reading my dissertation I hope it was as pleasant  
 0134 to read as it was to write I should elucidate that this is not a curse I quite enjoyed taking the time to put  
 0135 in writing what I learned over the last few years I shall also thank Carl Marletti creator of the Lily Valley  
 0136 with its indecently delicious or maybe deliciously indecent association of sugar fat and violets into a mean of  
 0137 reaching my goal of contracting diabetes on my deathbed note however that since it was contracted through  
 0138 the ingestion of violets it can surely be analyzed as poetical diabetes although still classified as type 2 Carl  
 0139 Your glory in what was an undeniable and mouthwatering emotional support shall not be forgotten I was  
 0140 furthermore corrupted into the sin of gluttony by Sadaharu Aoki Toque Cuivrée and Merveilleux de chez Fred  
 0141 I also enjoy Baillardran despite its overpricing and overtly bourgeois styling I should therefore not be too  
 0142 vocal about it in case liberation precedes alopecia I feel like I'm digressing about boltmakers again no if I  
 0143 have to thank someone for emotional support first and foremost that would be Raphaëlle Labarrière Syrielle  
 0144 Montariol and Billur Sezgin thank you for the numerous evenings and nights spent rebuilding the world and  
 0145 ourselves with an excessive amount of wine thank you for making me discover things that I enjoy about life  
 0146 in particular thank you for sharing a copious amount of literary works I particularly enjoy reading classical  
 0147 authors whose remains are in various states of decomposition between the retirement home and the Panthéon  
 0148 I had the pleasure of snatching several volumes from their skeletal hands thank you Raphaëlle for providing  
 0149 flawless directions in grave robbery I further thank Jill-Jënn Vie and once again Syrielle Montariol and Arij  
 0150 Riabi for proofreading my dissertation you did a great job correcting me and oh boy did I need it thankfully  
 0151 the first version was not witnessed by the official reviewers that's what's called taking one for the team next in  
 0152 order I would like to thank tomatoes in all their shapes and forms thank you for helping me keep a semblance  
 0153 of health for giving a bit more of a culinary personality beyond that of a sugar-addict even though I still have a  
 0154 kind of obsessive relationship with you I'm not ready to deconstruct it yet I wrote you a poem it goes like this  
 0155 Oh tomato Ô tomato 御tomato speaking of health thank you Dr Bouteille for taking care of me and having  
 0156 a funny name and now speaking of bottle I would like to thank sweet-tooth Syrielle Montariol and mixologist  
 0157 Kenza Jernite for being among the most admirable people I know the world is more kind with you in it I  
 0158 enjoyed sharing liquorous wine and cocktails with you and I have faith we will be getting diabetes together  
 0159 with moderation because on the other side Billur is accompanying me beyond moderation on cathartic disdain  
 0160 I can only hope we'll be getting to hell together but for now waiting for this unavoidable and deserved end  
 0161 I would like to thank my bed for being the nicest place on earth and not only my bed all the beds that  
 0162 sheltered me and those that will accommodate me in the future I shall even include couches or any surfaces

0163 appropriate for napping I long the days we were forced to sleep in kindergarten and I can only hope this will  
0164 be generalized to latter stages of life as I'm sleep-deprivedly writing these acknowledgements the night before  
0165 my defense you might think this clinophilia is revealing of an underlying pathology but fear not as I enjoy all  
0166 soft things made from fabric I take the opportunity to thank the kind overseeing of my work by communist  
0167 BERT and the transitional support of Waza Brick Oishii et al despite your stoic airs I know how you feel deep  
0168 inside I should also thank several members of the Felidae family for providing purring especially Ozwin you're  
0169 a coward but you're cute you're a cute coward you're worth your weight of catnip when you get the chance  
0170 of meeting her in a courageous disposition you can hold her little furry paws in your hand feel the softness  
0171 of this unique specimen yes at times though her dark side overwhelms her and wounds you while you're in  
0172 the most playful mood she nonchalantly write a Greek tragedy in the thunderous strength of her claws yes  
0173 yes but cutely though speaking of family I should thank mine which boringly is Hominidae the end of the  
0174 genealogy tree looks like  $\searrow \swarrow$  yes at depth 2 it's still a tree that was an intelligent decision by my ancestors  
0175 and though I don't remember everything I am thankful for your support early in life I would also like to thank  
0176 the action de groupe and its 群作用 extension in particular thank you Jill-Jënn for jill-jënnning all along thank  
0177 you Shlob for laughing at my poorly crafted puns or at anything really you're too good of a public thank  
0178 you Tito for indirectly teaching me more about machine learning than what you yourself know thank you  
0179 Alex for teaching me more about machine learning than what you wish you knew thank you Link Mauve for  
0180 showing me you can sluggishly not care about unimportant things thank you Ryan for the gentle squabble and  
0181 thank you again Tito Alex and Link Mauve for your substantiated subversion speaking of subversion I would  
0182 like to thank the whole open source community and more broadly organizations encouraging the sharing of  
0183 information with as many people as possible such as Wikipedia and Sci-Hub but I don't need to go this far for  
0184 finding people sharing ideas I would like to thank all members of the MLIA team for teaching me in particular  
0185 a huge thank to the people who participated in our reading groups that was legitimately the best work-related  
0186 moments during my time in the lab if we go into non-work-related we might end up in some incongruous  
0187 raclette-karaoke night I would also like to thank the people in the bestest office ever 26-00/534 thank you  
0188 Marie for your strong laughter Agnès for your strong chill Tristan for your strong flow and Jean-Yves for  
0189 your strong temporal consistency I'm sure the time at which you leave for your afternoon collation could have  
0190 been used to calibrate atomic clocks thus providing an unyielding beacon of stability in research's messy life  
0191 finally thank you Christophe Boudier for dealing with deep learning ludicrous computational requirements and  
0192 for providing a serious challenge in foosball this remind me that I should also thank my sport mates Syrielle  
0193 and 26-00/534 for bouldering and Arij for swimming a deep thanks to the municipal employees who decided  
0194 on the nearly 30 degrees Celsius temperature for the swimming pool I wonder whether you might have been  
0195 doing more for my physical health than a bottle lost in a municipal health center I take the opportunity to  
0196 thank the love of my life with whom I share everyday hot water I sing your praise everyday a big thank you  
0197 to Billur Sezgin for lending me her bathtub and thank you to Lush for achieving the feat of making it more  
0198 enjoyable thank you also to Manon Dumas Morès for accepting the same bathtub-lending deal I might become  
0199 a bathtub tycoon in the future I partly grew up in a thermal city maybe that explains why I like hot water so  
0200 much I'm not sure but just in case thank you to my thermal city Bagnères-de-Luchon to hell with it thank you  
0201 to all thermal cities they deserve it well maybe one instance of enjoyable cold water was a night bath in Kyoto  
0202 for Gozan no Okuribi thank you to the protagonist of this otherworldly night for making it so memorable the  
0203 cold water appeared first in the Takasegawa channel near Pontocho then in the duck's river Kamogawa which  
0204 despite its name was void of ducks which might have hidden following their awareness of a cooking intent  
0205 originating in a native of southwestern France thank you to Syrielle Kenza and Manon for participating in  
0206 singular culinary experiments this is what inspired the OuCuiPo illustration in my introduction sorry Manon  
0207 though for getting you sick with some weird black pepper ok it's getting late thank you Arthur Suspene for  
0208 getting old as fast as I do but slightly earlier thank you to Sappho's friend who shall sadly remain unnamed  
0209 welcome to Jill-Jënn's firework I'm sure it's going to turn out better than the movie and long life to Anne  
0210 Émone and all her children I want to leave her offspring in 26-00/534 so that we can share something beyond  
0211 our PhD but I fear most of them will starve to death by the end of the month finally thank you to all the  
0212 excellent teachers who accompanied me until the end of my formal studies I aspire to be half as good as you  
0213 were.  
0214  
0215  
0216

0217  
 0218  
 0219  
 0220  
 0221  
 0222  
 0223  
 0224  
 0225  
 0226  
 0227  
 0228  
 0229  
 0230  
 0231  
 0232  
 0233  
 0234  
 0235  
 0236  
 0237  
 0238  
 0239  
 0240  
 0241  
 0242  
 0243  
 0244  
 0245  
 0246  
 0247  
 0248  
 0249  
 0250  
 0251  
 0252  
 0253  
 0254  
 0255  
 0256  
 0257  
 0258  
 0259  
 0260  
 0261  
 0262  
 0263  
 0264  
 0265  
 0266  
 0267  
 0268  
 0269  
 0270

“Michael: Yes—it wasn’t logical.

George : You were a tomato! A tomato doesn’t have logic. A tomato can’t move.

—“Tootsie” (1982)

“This disaster of the Cherokees, brought to me by a sad friend to blacken my days and nights! I can do nothing; why shriek? why strike ineffectual blows? I stir in it for the sad reason that no other mortal will move, and if I do not, why, it is left undone. The amount of it, to be sure, is merely a scream; but sometimes a scream is better than a thesis.

—Ralph Waldo Emerson “Letter to President van Buren” (1838)

“Aaaaaaaaaah

—Alain Chabat in “Reality” by Quentin Dupieux (2014)

0271  
0272  
0273  
0274  
0275  
0276  
0277  
0278  
0279  
0280  
0281  
0282  
0283  
0284  
0285  
0286  
0287  
0288  
0289  
0290  
0291  
0292  
0293  
0294  
0295  
0296  
0297  
0298  
0299  
0300  
0301  
0302  
0303  
0304  
0305  
0306  
0307  
0308  
0309  
0310  
0311  
0312  
0313  
0314  
0315  
0316  
0317  
0318  
0319  
0320  
0321  
0322  
0323  
0324



0325		
0326		
0327		
0328		
0329		
0330	<b>Contents</b>	
0331		
0332		
0333		
0334		
0335		
0336	<b>Abstract</b>	<b>iii</b>
0337		
0338	<b>Acknowledgements</b>	<b>v</b>
0339		
0340	<b>List of Abbreviations</b>	<b>xv</b>
0341		
0342	<b>Notation</b>	<b>xvii</b>
0343		
0344	<b>Introduction</b>	<b>xix</b>
0345	<b>1 Context: Distributed Representations</b>	<b>25</b>
0346	1.1 Historical Development . . . . .	25
0347	1.2 Distributed Representation of Words . . . . .	28
0348	1.2.1 Word2vec . . . . .	29
0349	1.2.1.1 Skip-gram . . . . .	29
0350	1.2.1.2 Noise Contrastive Estimation . . . . .	29
0351	1.2.1.3 Negative Sampling . . . . .	30
0352	1.2.2 Language Modeling for Word Representation . . . . .	30
0353	1.2.3 Subword Tokens . . . . .	31
0354	1.3 Distributed Representation of Sentences . . . . .	32
0355	1.3.1 Convolutional Neural Network . . . . .	32
0356	1.3.2 Recurrent Neural Network . . . . .	33
0357	1.3.2.1 Long Short-term Memory . . . . .	33
0358	1.3.2.2 ELMO . . . . .	35
0359	1.3.3 Attention Mechanism . . . . .	35
0360	1.3.3.1 Attention as a Mechanism for RNN . . . . .	35
0361	1.3.3.2 Attention as a Standalone Model . . . . .	36
0362	1.3.4 Transformers . . . . .	37
0363	1.3.4.1 Transformer Attention . . . . .	37
0364	1.3.4.2 Masked Language Model . . . . .	38
0365	1.3.4.3 Transfer Learning . . . . .	38
0366	1.4 Knowledge Base . . . . .	39
0367	1.4.1 Relation Algebra . . . . .	40
0368	1.4.2 Distributed Representation through Knowledge Base Completion . . . . .	41
0369	1.4.2.1 Selectional Preferences . . . . .	42
0370	1.4.2.2 RESCAL . . . . .	42
0371	1.4.2.3 TransE . . . . .	43
0372	1.5 Conclusion . . . . .	44
0373		
0374	<b>2 Relation Extraction</b>	<b>47</b>
0375	2.1 Task Definitions . . . . .	48
0376	2.1.1 Nature of Relations . . . . .	51
0377	2.1.1.1 Unspecified Relation: <i>Other</i> . . . . .	51
0378	2.1.1.2 Closed-domain Assumption . . . . .	51

0379	2.1.1.3	Directionality and Ontology . . . . .	51
0380	2.1.2	Nature of Entities . . . . .	52
0381	2.2	The Problem of Data Scarcity . . . . .	53
0382	2.2.1	Bootstrap . . . . .	53
0383	2.2.2	Distant Supervision . . . . .	54
0384	2.3	Supervised Sentential Extraction Models . . . . .	55
0385	2.3.1	Evaluation . . . . .	55
0386	2.3.2	Regular Expressions: DIPRE . . . . .	57
0387	2.3.3	Dependency Trees: DIRT . . . . .	58
0388	2.3.4	Hand-designed Feature Extractors . . . . .	59
0389	2.3.5	Kernel Approaches . . . . .	61
0390	2.3.6	Piecewise Convolutional Neural Network . . . . .	61
0391	2.3.7	Transformer-based Models . . . . .	62
0392	2.4	Supervised Aggregate Extraction Models . . . . .	63
0393	2.4.1	Label Propagation . . . . .	63
0394	2.4.2	Multi-instance Multi-label . . . . .	64
0395	2.4.3	Universal Schemas . . . . .	66
0396	2.4.4	Aggregate PCNN Extraction . . . . .	67
0397	2.4.5	Entity Pair Graph . . . . .	68
0398	2.5	Unsupervised Extraction Models . . . . .	69
0399	2.5.1	Evaluation . . . . .	69
0400	2.5.1.1	Clustering Metrics . . . . .	70
0401	2.5.1.2	Few-shot . . . . .	72
0402	2.5.2	Open Information Extraction . . . . .	73
0403	2.5.3	Clustering Surface Forms . . . . .	74
0404	2.5.4	Rel-LDA . . . . .	75
0405	2.5.5	Variational Autoencoder for Relation Extraction . . . . .	76
0406	2.5.6	Matching the Blanks . . . . .	78
0407	2.5.7	SelfORE . . . . .	80
0408	2.6	Conclusion . . . . .	81
0409			
0410	<b>3</b>	<b>Regularizing Discriminative Unsupervised Relation Extraction Models</b>	<b>83</b>
0411	3.1	Model description . . . . .	84
0412	3.1.1	Unsupervised Relation Classifier . . . . .	85
0413	3.1.2	Entity Predictor . . . . .	86
0414	3.1.3	RelDist losses . . . . .	88
0415	3.2	Related Work . . . . .	89
0416	3.3	Experiments . . . . .	90
0417	3.3.1	Datasets . . . . .	90
0418	3.3.2	Baselines and Models . . . . .	91
0419	3.3.3	Results . . . . .	92
0420	3.3.4	Qualitative Analysis . . . . .	94
0421	3.4	Alternative Models . . . . .	95
0422	3.5	Conclusion . . . . .	97
0423			
0424	<b>4</b>	<b>Graph-Based Aggregate Modeling</b>	<b>99</b>
0425	4.1	Encoding Relation Extraction as a Graph Problem . . . . .	100
0426	4.2	Preliminary Analysis and Proof of Principle . . . . .	102
0427	4.3	Related Work . . . . .	105
0428	4.3.1	Random-Walk-Based Models . . . . .	106
0429	4.3.2	Spectral GCN . . . . .	107
0430	4.3.3	Spatial GCN . . . . .	109
0431	4.3.4	GCN on Relation Graphs . . . . .	111
0432	4.3.5	Weisfeiler–Leman Isomorphism Test . . . . .	112

0433	4.4	Proposed Approaches . . . . .	114
0434	4.4.1	Using Topological Features . . . . .	114
0435	4.4.2	Nonparametric Weisfeiler–Leman Iterations . . . . .	115
0436	4.4.3	Refining Linguistic and Topological Features . . . . .	117
0437	4.5	Experiments . . . . .	118
0438	4.6	Conclusion . . . . .	119
0439			
0440		<b>Conclusion</b>	<b>121</b>
0441			
0442		<b>A Résumé en français</b>	<b>125</b>
0443	A.1	Contexte . . . . .	126
0444	A.2	Régularisation des modèles discriminatifs d’extraction non supervisée de relations . . . . .	129
0445	A.3	Modélisation à l’aide de graphes de la structure des jeux de données . . . . .	130
0446	A.4	Conclusion . . . . .	132
0447			
0448		<b>B List of Assumptions</b>	<b>133</b>
0449			
0450		<b>C Datasets</b>	<b>137</b>
0451	C.1	ACE . . . . .	137
0452	C.2	FewRel . . . . .	137
0453	C.3	Freebase . . . . .	138
0454	C.4	MUC-7 TR . . . . .	139
0455	C.5	New York Times . . . . .	139
0456	C.6	SemEval 2010 Task 8 . . . . .	139
0457	C.7	T-REX . . . . .	140
0458	C.8	Wikidata . . . . .	140
0459			
0460		<b>Bibliography</b>	<b>143</b>
0461			
0462			
0463			
0464			

## List of Figures

0470	1.1	Word2vec embeddings PCA. . . . .	28
0471	1.2	Architecture of a single convolutional filter with a pooling layer. . . . .	32
0472	1.3	RNN language model unrolled through time. . . . .	33
0473	1.4	Architecture of an LSTM cell. . . . .	34
0474	1.5	Schema of an attention mechanism. . . . .	36
0475	1.6	Schema of a memory network language model with two layers. . . . .	36
0476	1.7	Schema of BERT, a transformer masked language model. . . . .	38
0477	1.8	Structure of a knowledge base fact. . . . .	39
0478			
0479	2.1	The three standard tasks for knowledge base population. . . . .	48
0480	2.2	Supervised metrics defined on the confusion matrix. . . . .	57
0481	2.3	DIPRE split-in-three-affixes method. . . . .	57
0482	2.4	Example of dependency tree. . . . .	58
0483	2.5	Example of syntactic parse tree. . . . .	60
0484	2.6	Architecture of a PCNN model. . . . .	62
0485	2.7	MTB entity markers–entity start sentence representation. . . . .	63
0486	2.8	Multi-instance multi-label (MIML) setup. . . . .	65

0487	2.9	MultiR plate diagram. . . . .	65
0488	2.10	Universal schema matrix. . . . .	66
0489	2.11	Entity pair graph. . . . .	68
0490	2.12	EPGNN sentence representation. . . . .	69
0491	2.13	Comparison of $B^3$ and V-measure. . . . .	71
0492	2.14	Rel-LDA plate diagram. . . . .	76
0493	2.15	VAE plate diagram. . . . .	77
0494	2.16	Marcheggiani and Titov (2016) plate diagram. . . . .	77
0495	2.17	SelfORE iterative algorithm. . . . .	80
0496			
0497	3.1	Fill-in-the-blanks sentence partition. . . . .	85
0498	3.2	Illustration of $\mathcal{P}1$ . . . . .	88
0499	3.3	Illustration of $\mathcal{P}2$ . . . . .	88
0500	3.4	Confusion matrices on the T-REX SPO dataset. . . . .	94
0501			
0502	4.1	Multigraph construction example. . . . .	101
0503	4.2	T-REX vertices degree distribution. . . . .	102
0504	4.3	NELL dataset bipartite graph. . . . .	107
0505	4.4	Parallel between two-dimensional CNN data and GCN data. . . . .	109
0506	4.5	Example of isomorphic graphs. . . . .	112
0507	4.6	Example of line graph construction. . . . .	115
0508			
0509	A.1	Illustration du problème d'uniformité. . . . .	130
0510	A.2	Exemple de graphes isomorphes. . . . .	131
0511	A.3	Schéma de BERT, un modèle de langue masqué basé sur un <i>transformer</i> . . . . .	131
0512			
0513	C.1	Structure of a Wikidata page. . . . .	141
0514			
0515			
0516			
0517			
0518			
0519			
0520			
0521			
0522			
0523	1.1	Relation properties expressed in relation algebra. . . . .	41
0524			
0525	2.1	Example of supervised samples from the FewRel dataset . . . . .	50
0526	2.2	Few-shot problem. . . . .	72
0527			
0528	3.1	Quantitative results of clustering models. . . . .	93
0529	3.2	Quantitative results of the Gumbel-Softmax model on the NYT + FB dataset. . . . .	96
0530	3.3	Quantitative results of the alignment models on the NYT + FB dataset. . . . .	97
0531			
0532	4.1	Frequencies of some paths of length 3 in T-REX. . . . .	105
0533	4.2	Preliminary results for FewRel valid accuracies of graph-based approaches. . . . .	119
0534			
0535	A.1	Résultats quantitatifs des méthodes de partitionnement sur le dataset NYT-FB. . . . .	130
0536	A.2	Résultats quantitatifs des méthodes à base de graphe sur le jeu de données FewRel. . . . .	132
0537			
0538	C.1	Statistics of the FewRel dataset. . . . .	138
0539	C.2	Statistics of the Freebase knowledge base. . . . .	138
0540	C.3	Statistics of the SemEval 2010 Task 8 dataset. . . . .	140
	C.4	Statistics of the T-REX dataset. . . . .	140

## List of Tables

0541  
 0542  
 0543  
 0544  
 0545  
 0546  
 0547  
 0548  
 0549  
 0550  
 0551  
 0552  
 0553  
 0554  
 0555  
 0556  
 0557  
 0558  
 0559  
 0560  
 0561  
 0562  
 0563  
 0564  
 0565  
 0566  
 0567  
 0568  
 0569  
 0570  
 0571  
 0572  
 0573  
 0574  
 0575  
 0576  
 0577  
 0578  
 0579  
 0580  
 0581  
 0582  
 0583  
 0584  
 0585  
 0586  
 0587  
 0588  
 0589  
 0590  
 0591  
 0592  
 0593  
 0594

# List of Algorithms

1.1	The byte pair encoding algorithm. . . . .	32
1.2	The TransE training algorithm. . . . .	44
2.1	The bootstrap algorithm. . . . .	53
2.2	The label propagation algorithm. . . . .	64
2.3	The MultiR training algorithm. . . . .	66
2.4	The rel-LDA generative process. . . . .	76
4.1	Path counting algorithm . . . . .	104
4.2	The Weisfeiler–Leman isomorphism test. . . . .	113

0595  
0596  
0597  
0598  
0599  
0600  
0601  
0602  
0603  
0604  
0605  
0606  
0607  
0608  
0609  
0610  
0611  
0612  
0613  
0614  
0615  
0616  
0617  
0618  
0619  
0620  
0621  
0622  
0623  
0624  
0625  
0626  
0627  
0628  
0629  
0630  
0631  
0632  
0633  
0634  
0635  
0636  
0637  
0638  
0639  
0640  
0641  
0642  
0643  
0644  
0645  
0646  
0647  
0648

0649  
 0650  
 0651  
 0652  
 0653  
 0654  
 0655  
 0656  
 0657  
 0658  
 0659  
 0660  
 0661  
 0662  
 0663  
 0664  
 0665  
 0666  
 0667  
 0668  
 0669  
 0670  
 0671  
 0672  
 0673  
 0674  
 0675  
 0676  
 0677  
 0678  
 0679  
 0680  
 0681  
 0682  
 0683  
 0684  
 0685  
 0686  
 0687  
 0688  
 0689  
 0690  
 0691  
 0692  
 0693  
 0694  
 0695  
 0696  
 0697  
 0698  
 0699  
 0700  
 0701  
 0702

## List of Abbreviations

ACE	Automatic Content Extraction (Section C.1)
ACL	Association for Computational Linguistics
ARI	Adjusted Rand Index (Section 2.5.1.1)
BERT	Bidirectional Encoder Representations from Transformers (Section 1.3.4)
BPE	Byte-Pair Encoding (Section 1.2.3)
BPR	Bayesian Personalized Ranking (Section 2.4.3)
CNN	Convolutional Neural Network (Section 1.3.1)
DAE	Denoising AutoEncoder (Section 2.5.7)
DARPA	Defense Advanced Research Projects Agency (Section C.4)
DEC	Deep Embedded Clustering (Section 2.5.7)
DIPRE	Dual Iterative Pattern Relation Expansion (Section 2.3.2)
DIRT	Discovery of Inference Rules from Text (Section 2.3.3)
ELBO	Evidence Lower BOund (Section 2.5.5)
ELMO	Embeddings from Language Model (Section 1.3.2.2)
EPGNN	Entity Pair Graph Neural Network (Section 2.4.5)
FB	FreeBase (Section C.3)
GAT	Graph ATtention network (Section 4.3.3)
GCN	Graph Convolutional Network (Section 4.3)
GNN	Graph Neural Network (Section 4.3)
GIS	Generalized Iterative Scaling (Section 2.3.4)
GPE	Geo-Political Entity (Section 2.5.3)
GRU	Gated Recurrent Unit (Section 1.3.2.1)
IDF	Inverse Document Frequency (Section 2.5.3)
JSD	Jensen–Shannon Divergence (Section 3.4)
LDA	Latent Dirichlet Allocation (Section 2.5.4)
LSA	Latent Semantic Analysis (Section 1.2)
LSI	Latent Semantic Indexing (Section 1.2)
LSTM	Long Short-Term Memory (Section 1.3.2.1)
MIML	Multi-Instance Multi-Label (Section 2.4.2)
MLM	Masked Language Model (Section 1.3.4.2)
MTB	Matching The Blanks (Sections 2.3.7 and 2.5.6)
MUC	Message Understanding Conference (Section C.4)
NLP	Natural Language Processing (Sections 1.2 and 1.3)
NCE	Noise Contrastive Estimation (Section 1.2.1.2)
NER	Named Entity Recognition (Chapter 2)
NIST	National Institute of Standards and Technology (Section C.1)
NMT	Neural Machine Translation (Section 1.3.3)
NYT	New York Times (Section C.5)
OIE	Open Information Extraction (Section 2.5.2)
PCNN	Piecewise Convolutional Neural Network (Section 2.3.6)
PMI	Pointwise Mutual Information (Section 2.3.3)

0703	POS	Part Of Speech (Figure 2.4)
0704	RI	Rand Index (Section 2.5.1.1)
0705	RNN	Recurrent Neural Network (Section 1.3.2)
0706	SVM	Support Vector Machine (Section 2.3.5)
0707	SGNS	Skip-Gram Negative Sampling (Section 1.2.1)
0708	TF	Term Frequency (Section 2.5.3)
0709	VAE	Variational AutoEncoder (Section 2.5.5)
0710	WL	Weisfeiler–Leman isomorphism test (Section 4.3.5)
0711	WMT	Workshop on statistical Machine Translation (Section 1.1)
0712		
0713		
0714		
0715		
0716		
0717		
0718		
0719		
0720		
0721		
0722		
0723		
0724		
0725		
0726		
0727		
0728		
0729		
0730		
0731		
0732		
0733		
0734		
0735		
0736		
0737		
0738		
0739		
0740		
0741		
0742		
0743		
0744		
0745		
0746		
0747		
0748		
0749		
0750		
0751		
0752		
0753		
0754		
0755		
0756		



0757  
 0758  
 0759  
 0760  
 0761  
 0762  
 0763  
 0764  
 0765  
 0766  
 0767  
 0768  
 0769  
 0770  
 0771  
 0772  
 0773  
 0774  
 0775  
 0776  
 0777  
 0778  
 0779  
 0780  
 0781  
 0782  
 0783  
 0784  
 0785  
 0786  
 0787  
 0788  
 0789  
 0790  
 0791  
 0792  
 0793  
 0794  
 0795  
 0796  
 0797  
 0798  
 0799  
 0800  
 0801  
 0802  
 0803  
 0804  
 0805  
 0806  
 0807  
 0808  
 0809  
 0810

## Notation

Most of this thesis is formatted in one and a half columns, which means that a large right margin is filled with complementary material. This includes figures, tables and algorithms when space allows, but also epigraphs and marginal notes with supplementary details and comments. The titles of important bibliographical references are also given in the margin right of their first mention in the section. Some marginal paragraphs are left unnumbered and provide material about the broadly adjacent passage. When a section seems unclear, we invite the reader to look for additional information in the margin. For example, while relation algebra is introduced in Section 1.4.1, we do not expect most readers to be familiar with its notation. As such, we will systematically provide an interpretation of relation algebra formulae in plain English in unnumbered marginal paragraphs.

### Domain of Variables

$x$	A scalar
$\mathbf{x}$	A vector, its elements are indexed $x_i$
$\mathbf{X}$	A matrix, its rows are indexed $\mathbf{x}_i$ , its elements $x_{ij}$
$\mathbf{X}$	A (three-way) tensor, indexed $\mathbf{X}_i, \mathbf{x}_{ij}, x_{ijk}$
$x$	A random variable (sometimes $X$ to avoid confusion)
$\mathbf{x}$	A random vector
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	The set of real-valued vectors of length $n$
$\mathbb{R}^{n \times m}$	The set of real-valued matrices with $n$ rows and $m$ columns
$B^A$	The set of functions from $A$ to $B$ , in particular $2^A$ denotes the power set of $A$

To describe the set of real-valued vectors with the same number of elements as a set  $A$ , we abuse the morphism from the functions  $\mathbb{R}^A$  to the vectors  $\mathbb{R}^{|A|}$  and simply write  $\mathbf{x} \in \mathbb{R}^A$  to denote that  $\mathbf{x}$  is a vector with  $|A|$  elements.

### Relation Algebra

Relation algebra is described in more detail in Section 1.4.1.

$\mathbf{0}$	Empty relation
$\mathbf{1}$	Complete relation
$\mathbf{I}$	Identity relation
$\bar{r}$	Complementary relation
$\check{r}$	Converse relation (reversed orientation), when applied to a surface form: $\overline{\text{born in}}$
$\bullet$	Relation composition

### Probability and Information Theory

$P(x), Q(x)$	Probability distribution over $x$ , by default we heavily overload $P$ (as is customary), when confusion is possible we disambiguate by using $Q$
$\hat{P}(x)$	Empirical distribution over $x$ (as defined by the dataset)
$x \perp\!\!\!\perp y \mid z$	Conditional independence of $x$ and $y$ given $z$
$x \not\perp\!\!\!\perp y$	$x$ and $y$ are not independent
$\mathcal{U}(X)$	Uniform distribution over the set $X$

0811	$\mathcal{N}(\mu, \sigma^2)$	Normal distribution of mean $\mu$ and variance $\sigma^2$ (also used for the multivariate case)
0812	$H(x)$	Shannon entropy of the random variable $x$ , $H(x, y)$ denotes the joint entropy
0813	$H(x   y)$	Conditional entropy of $x$ given $y$
0814	$H_Q(P)$	Cross-entropy of $P$ relative to $Q$
0815	$I(x; y)$	Mutual information of $x$ and $y$
0816	$\text{pmi}(x, y)$	Pointwise mutual information of events $x$ and $y$
0817	$D_{\text{KL}}(P \  Q)$	Kullback–Leibler divergence from $Q$ to $P$
0818	$D_{\text{JS}}(P \  Q)$	Jensen–Shannon divergence between $P$ and $Q$
0819	$W_1(P, Q)$	1-Wasserstein distance between $P$ and $Q$
0820		
0821		<b>Machine Learning</b>
0822	$\sigma(x)$	Logistic sigmoid $\sigma(x) = 1 / (1 + \exp(-x))$
0823	$\text{ReLU}(x)$	Rectified linear unit $\text{ReLU}(x) = \max(0, x)$ , we use $\text{ReLU}_{\bullet}$ to refer to the ReLU activation applied to half of the units (see Section 1.3.3.2)
0824		
0825	$\mathcal{L}$	Loss (to be minimized)
0826	$J$	Objective (to be maximized)
0827	$\vec{F}_1, \overleftarrow{F}_1, \overleftarrow{F}_1$	Directed, undirected and half-directed $F_1$ measures (see Section 2.3.1)
0828		
0829		<b>Graph Operations</b>
0830		
0831	$\varepsilon_1(a)$	Source vertex of the arc $a$
0832	$\varepsilon_2(a)$	Target vertex of the arc $a$
0833	$\rho(a)$	Relation conveyed by the arc $a$
0834	$\varsigma(a)$	Sentence corresponding to the arc $a$
0835	$N(e)$	Vertices neighboring the vertex $e$
0836	$\mathcal{I}(e)$	Arcs incident to the vertex $e$
0837	$\mathcal{N}(a)$	Arcs neighboring the arc $a$
0838		
0839		<b>Other Operations</b>
0840	$\odot$	Element-wise (Hadamard) product
0841	$*$	Convolution
0842	$\bowtie$	Natural join
0843	$\times_A$	Pullback with common codomain $A$
0844	$\delta_{i,j}$	Kronecker’s delta, 1 if $i = j$ , 0 otherwise
0845		
0846		
0847		
0848		
0849		
0850		
0851		
0852		
0853		
0854		
0855		
0856		
0857		
0858		
0859		
0860		
0861		
0862		
0863		
0864		

# Introduction

The world is endowed with a structure, which enables us to understand it. This structure is most apparent through repetitions of sensory experiences. Sometimes, we can see a cat, then another cat. Entities emerge from the repetition of catness we experienced. From time to time, we can also observe a cat *inside* a cardboard box or a person *inside* a room. Relations are the explanatory device underlying this second kind of repetition. A relation governs an interaction between two or more objects. We assume an *inside* relation exists because we repeatedly experienced the same interaction between a container and its content. The twentieth century saw the rise of structuralism, which regarded the interrelations of phenomena as more enlightening than the study of phenomena in isolation. In other words, we might better understand what a cat is by studying its relationships to other entities instead than by listing the characteristics of catness. From this point of view, the concept of relation is crucial to our understanding of the world.

Natural languages capture the underlying structure of these repetitions through a process we do not fully understand. One of the endeavors of artificial intelligence, called natural-language understanding, is to mimic this process with definite algorithms. Since the aforementioned goal is still elusive, we strive to model only parts of this process. This thesis, consequent to the structuralist perspective, focuses on extracting relations conveyed by natural language. Assuming natural language is representative of the underlying structure of sensory experiences,<sup>1</sup> we should be able to capture relations through the exploitation of repetitions alone—i.e. in an unsupervised fashion.

Extracting relations can help better our understanding of how languages work. For example, whether languages can be understood through a small amount of data is still a somewhat open question in linguistics. The poverty of the stimulus argument states that children should not be able to acquire proficiency from being exposed to so little data. It is one of the major arguments in favor of the controversial universal grammar theory. Capturing relations from nothing more than a small number of natural language utterances would be a step towards disproving the poverty of the stimulus claim.

Relations—albeit in a more restrictive sense—are one of Aristotle’s ten *praedicamenta*, the categories of objects of human apprehension (Gracia and Newton 2016).



The Cheshire Cat from Tenniel (1889) provides you with an experience of catness.

<sup>1</sup> The repetitions of sensory experiences and words need not be alike. We are only concerned with the possibility of resolving references here. Even though our experiences of trees are more often than not accompanied with experiences of bark, the words “tree” and “bark” do not co-occur as often in natural language utterances. However, their meronymic relationship is understandable both through experiences of trees and inter alia through the use of the preposition “of” in textual mentions of barks.

0919 This kind of incentive for tackling the relation extraction problem  
 0920 stems from an *episteme*<sup>2</sup> endeavor. However, most of the traction for this  
 0921 problem stems from a *techne*<sup>3</sup> undertaking. The end goal is to build a  
 0922 system with real-world applications. Under this perspective, the point of  
 0923 artificial intelligence is to replace or assist humans on specific tasks. Most  
 0924 tasks of interest necessitate some form of technical knowledge (e.g. diag-  
 0925 nosing a disease requires knowledge of the relationship between symptoms  
 0926 and diseases). The principal vector of knowledge is language (e.g. through  
 0927 education). Thus, knowledge acquisition from natural language is funda-  
 0928 mental for systems purposing to have such applications.  
 0930

0931 For an analysis of the real-world impact of systems extracting knowl-  
 0932 edge from text, refer to Alex et al. (2008). Their article shows that human  
 0933 curators can use a machine learning system to better extract a set of  
 0934 protein–protein interactions from biomedical literature. This is clearly a  
 0935 *techne* endeavor: the protein–protein interactions are not new knowledge,  
 0936 they are already published; however, the system improves the work of the  
 0937 human operator.  
 0939

0940 This example of application is revealing of the larger problem of infor-  
 0941 mation explosion. The quantity of published information has grown relent-  
 0942 lessly throughout the last decades. Machine learning can be used to filter  
 0943 or aggregate this large amount of data. In this case, the object of interest  
 0944 is not the text in itself but the conveyed semantic, its meaning. This begs  
 0945 the question: how to define the meaning we are seeking to process? Indeed,  
 0946 foundational theories of meaning are the object of much discussion in the  
 0947 philosophy community (Speaks 2021). While some skeptics, like Quine, do  
 0948 not recognize meaning as a concept of interest, they reckon that a mini-  
 0949 mal description of meaning should at least encompass the recognition of  
 0950 synonymy. This follows from the above discussion about the recognition of  
 0951 repetitions: if  $\clubsuit$  is a repetition of  $\spadesuit$ , we should be able to say that  $\clubsuit$  and  
 0952  $\spadesuit$  are synonymous. In practice, this implies that we ought to be able to  
 0953 extract classes of linguistic forms with the same meaning or referent—the  
 0954 difference between the two is not relevant to our problem.  
 0958

0959 While the above discussion of meaning is essential to define our objects  
 0960 of interest, relations, it is important to note that we work on language; we  
 0961 want to extract relations from language, not from repetitions of abstract  
 0962 entities. Yet, the mapping between linguistic signifiers and their meaning  
 0963 is not bijective. We can distinguish two kinds of misalignment between  
 0964 the two: either two expressions refer to the same object (synonymy), or  
 0965 the same expression refers to different objects depending on the context  
 0966 in which it appears (homonymy). The first variety of misalignment is the  
 0967 most common one, especially at the sentence level. For example, “Paris is  
 0968 the capital of France” and “the capital of France is Paris” convey the same  
 0969 meaning despite having different written and spoken forms. On the other  
 0970  
 0971  
 0972

<sup>2</sup> From the Ancient Greek ἐπιστήμη: knowledge, know-how.

<sup>3</sup> From the Ancient Greek τέχνη: craft, art.

Alex et al., “Assisted curation: does text mining really help?” PSB 2008

“Once the theory of meaning is sharply separated from the theory of reference, it is a short step to recognizing as the business of the theory of meaning simply the synonymy of linguistic forms and the analyticity of statements; meanings themselves, as obscure intermediary entities, may well be abandoned.

— Willard Van Orman Quine, “Main Trends in Recent Philosophy: Two Dogmas of Empiricism” (1951)



Paris (Q162121) is neither capital of France, nor prince of Troy, it is the genus of the true lover’s knot plant. The capital of France would be Paris (Q90) and the prince of Troy, son of Priam, Paris (Q167646). Illustration from Redouté (1802).

0973 hand, the second kind is principally visible at the word level. For example,  
 0974 the preposition “from” in the phrases “retinopathy from diabetes” and  
 0975 “Bellerophon from Corinth” conveys either a *has effect* relationship or  
 0976 a *birthplace* one. To distinguish these two uses of “from,” we can use  
 0977 relation identifiers such as P1542 for *has effect* and P19 for *birthplace*. An  
 0978 example with entity identifiers—which purpose to uniquely identify entity  
 0979 concepts—is provided in the margin of page xx.

0981 While the preceding discussion makes it seem as if all objects can  
 0982 fit nicely into clearly defined concepts, in practice, this is far from the  
 0983 truth. Early in the knowledge-representation literature, Brachman (1983)  
 0984 remarked the difficulty to clearly define even seemingly simple relations  
 0985 such as *instance of* (P31). This problem ensues from the assumption that  
 0986 synonymy is transitive, and therefore, induces equivalence classes. This  
 0987 assumption is fairly natural since it already applies to the link between  
 0988 language and its references: even though two cats might be very unlike  
 0989 one another, we still group them under the same signifier. However, lan-  
 0990 guage is flexible. When trying to capture the entity “cat,” it is not entirely  
 0991 clear whether we should group “a cat with the body of a cherry pop tart”  
 0992 with regular experiences of catness.<sup>4</sup> To circumvent this issue, some re-  
 0993 cent works (Han et al. 2018) on the relation extraction problem define  
 0994 synonymy as a continuous intransitive association. Instead of grouping  
 0995 linguistic forms into clear-cut classes with a single meaning, they extract  
 0996 a similarity function defining how similar two objects are.

1001 Now that we have conceptualized our problem, let us focus on our  
 1002 proposed technical approach. First, to summarize, this thesis focus on  
 1003 unsupervised relation extraction from text.<sup>5</sup> Since relations are objects  
 1004 capturing the interactions between entities, our task is to find the relation  
 1005 linking two given entities in a piece of text. For example, in the three  
 1006 following samples where entities are underlined:

1009 Megrez<sub>e<sub>1</sub></sub> is a star in the northern circumpolar constellation  
 1010 of Ursa Major<sub>e<sub>2</sub></sub>.

1012 Posidonius<sub>e<sub>1</sub></sub> was a Greek philosopher, astronomer, historian,  
 1013 mathematician, and teacher native to Apamea, Syria<sub>e<sub>2</sub></sub>.

1015 Hipparchus<sub>e<sub>1</sub></sub> was born in Nicaea, Bithynia<sub>e<sub>2</sub></sub>, and probably  
 1016 died on the island of Rhodes, Greece.

1018 we wish to find that the last two sentences convey the same relation—in  
 1019 this case, *e<sub>1</sub> born in e<sub>2</sub>* (P19)—or at the very least, following the discussion  
 1020 in the preceding paragraph about the difficulty of defining clear relation  
 1021 classes, we wish to find that the relations conveyed by the last two sam-  
 1022 ples are closer to each other than the one conveyed by the first sample. We  
 1023 propound that this can be performed by machine learning algorithms. In  
 1024 particular, we study how to approach this task using deep learning. While  
 1025  
 1026

Throughout this thesis, we will be using Wikidata identifiers (<https://www.wikidata.org>) to index entities and relations. Entities identifiers start with Q, while relation identifiers start with P. For example, Q35120 is an entity.

<sup>4</sup> The reader who would describe this as a cat is invited to replace various body parts of this imaginary cat with food items until they stop experiencing catness.

<sup>5</sup> We use text as it is the most definite and easy-to-process rendition of language.



Ariadne waking on the shore of Naxos where she was abandoned, wall painting from Herculaneum in the collection of the British Museum (100 BCE–100 CE). The ship in the distance can be identified as the ship of Theseus, for now. Depending on the philosophical view of the reader (Q1050837), its identity as the ship of Theseus might not linger for long.

1027 relation extraction can be tackled as a standard supervised classification  
 1028 problem, labeling a dataset with precise relations is a tedious task, espe-  
 1029 cially with technical documents such as the biomedical literature studied  
 1030 by Alex et al. (2008). Another problem commonly encountered by anno-  
 1031 tators is the question of applicability of a relation, for example, should  
 1032 “the country<sub>*e*<sub>1</sub></sub>’s founding father<sub>*e*<sub>2</sub></sub>” be labeled with the *product–producer*  
 1033 relation?<sup>6</sup> We now discuss how deep learning became the most promising  
 1034 technique to tackle natural language processing problems.  
 1035  
 1036

1037 The primary subject matter of the relation extraction problem is lan-  
 1038 guage. Natural language processing (NLP) was already a prominent re-  
 1039 search interest in the early years of artificial intelligence. This can be seen  
 1040 from the *episteme* viewpoint in the seminal paper of Turing (1950). This  
 1041 paper proposes mastery of language as evidence of intelligence, in what is  
 1042 now known as the Turing test. Language was also a subject of interest for  
 1043 *techne* objectives. In January 1954, the Georgetown–IBM experiment tried  
 1044 to demonstrate the possibility of translating Russian into English using  
 1045 computers (Dostert 1955). The experiment showcased the translation of  
 1046 sixty sentences using a bilingual dictionary to translate words individu-  
 1047 ally and six kinds of grammatical rules to reorder tokens as needed. Initial  
 1048 experiments created an expectation buildup, which was followed by an un-  
 1049 avoidable disappointment, resulting in an “AI winter” where research fund-  
 1050 ings were restricted. While translating word-by-word is somewhat easy in  
 1051 most cases, translating whole sentences is a lot harder. Scaling up the set  
 1052 of grammatical rules in the Georgetown–IBM experiment proved imprac-  
 1053 tical. This limitation was not a technical one. With the improvement of  
 1054 computing machinery, more rules could have easily been encoded. One of  
 1055 the issues identified at the time was the commonsense knowledge problem  
 1056 (McCarthy 1959). In order to translate or, more generally, process a sen-  
 1057 tence, it needs to be understood in the context of the world in which it  
 1058 was uttered. Simple rewriting rules cannot capture this process.<sup>7</sup> In order  
 1059 to handle whole sentences, a paradigm shift was necessary.  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065

1066 A first shift occurred in the 1990s with the advent of statistical NLP  
 1067 (S. Abney 1996). This evolution can be partly attributed to the increase of  
 1068 computational power, but also to the progressive abandon of essentialist  
 1069 linguistics precepts<sup>8</sup> in favor of distributionalist ones. Instead of relying on  
 1070 human experts to input a set of rules, statistical approaches leveraged the  
 1071 repetitions in large text corpora to infer these rules automatically. There-  
 1072 fore, this progression can also be seen as a transition away from symbolic  
 1073 artificial intelligence models and towards statistical ones. Coincidentally, the  
 1074 relation extraction task was formalized at this time. And while the ear-  
 1075 liest approaches were based on symbolic models using handwritten rules,  
 1076 statistical methods quickly became the norm after the 1990s. However,  
 1077 statistical NLP models still relied on linguistic knowledge. The relation  
 1078  
 1079  
 1080

<sup>6</sup> The annotator of this sentence piece in the SemEval 2010 Task 8 dataset (Section C.6) decided that it does convey the *product–producer* relation. The difficulty of applying a definition is an additional argument in favor of similarity-function-based approaches over classification approaches.

Turing, “Computing Machinery and Intelligence” *Mind* 1950

“Five, perhaps three years hence, interlingual meaning conversion by electronic process in important functional areas of several languages may well be an accomplished fact.

— Leon Dostert, “701 translator”  
 IBM press release (1954)

<sup>7</sup> Furthermore, grammar is still an active area of research. We do not perfectly understand the underlying reality captured by most words and are thus unable to write down complete formal rules for their usages. For example, Tyler and Evans (2001) is a 43 pages cognitive linguistics paper attempting to explain the various uses of the English preposition “over.” This is one of the arguments for unsupervised approaches; we should avoid hand-labeled datasets if we want to outperform the human annotators.

<sup>8</sup> Noam Chomsky, one of the most—if not the most—prominent essentialist linguists, considers that manipulating probabilities of text excerpt is not the way to acquire a better understanding of language. Following the success of statistical approaches, he only recognized statistical NLP as a *techne* achievement. For an answer to this position, see S. Abney (1996) and Norvig (2011).

1081 extraction systems were usually split into a first phase of hand-specified  
 1082 linguistic features extraction and a second phase where a relation was  
 1083 predicted based on these features using shallow statistical models.

1084 A second shift occurred in the 2010s when deep learning approaches  
 1085 erased the split between feature extraction and prediction. Deep learning  
 1086 models are trained to directly process raw data, in our case text excerpts.  
 1087 To achieve this feat, neural networks able to approximate any function are  
 1088 used. However, the downside of these models is that they usually require  
 1089 large amounts of labeled data to be trained. This is a particularly salient  
 1090 problem throughout this thesis since we deal with an unsupervised problem.  
 1091 As the latest and most efficient technique available, deep learning  
 1092 proved to be a natural choice to tackle relation extraction. However, this  
 1093 natural evolution came with serious complications that we try to address  
 1094 in this manuscript.

1098 The evolution of unsupervised relation extraction methods closely fol-  
 1099 lows the one of NLP methods described above. The first deep learning ap-  
 1100 proach was the one of Marcheggiani and Titov (2016). However, only part  
 1101 of their model relied on deep learning techniques, the extraction of features  
 1102 was still done manually. The reason why feature extraction could not be  
 1103 done automatically as is standard in deep learning approaches is closely  
 1104 related to the unsupervised nature of the problem. Our first contribution  
 1105 is to propose a technique to enable the training of unsupervised fully-  
 1106 deep learning relation extraction approaches. Afterward, different ways  
 1107 to tackle the relation extraction task emerged. First, recent approaches  
 1108 use a softer definition of relations by extracting a similarity function in-  
 1109 stead of a classifier. Second, they consider a broader context: instead of  
 1110 processing each sentence individually, the global consistency of extracted  
 1111 relations is considered. However, this second approach was mostly limited  
 1112 to the supervised setting, with limited use in the unsupervised setting. Our  
 1113 second contribution concerns using this broader context for unsupervised  
 1114 relation extraction, in particular for approaches defining a similarity func-  
 1115 tion. During the preparation of the thesis, we also published an article on  
 1116 multimodal semantic role labeling with Syrielle Montariol and her team  
 1117 (Montariol et al. 2022); since it is somewhat unrelated to unsupervised  
 1118 relation extraction, we do not include it in this thesis.

1125 We now describe the organization of the thesis. Chapter 1 provides  
 1126 the necessary background for using deep learning to tackle the relation  
 1127 extraction problem. In particular, we focus on the concept of distributed  
 1128 representation, first of language, then of entities and relations. Chapter 2  
 1129 formalizes the relation extraction task and presents the evaluation frame-  
 1130 work and relevant related works. This chapter focuses first on supervised  
 1131 relation extraction using local information only, then on aggregate extrac-  
 1132 tion, which exploits repetitions more directly, before delving into unsu-  
 1133  
 1134

“White horse is not horse.  
 — “Gongsun Longzi” Chapter 2 (circa 300 BCE)

「白馬非馬」

A well-known paradox in early Chinese philosophy illustrating the difficulty of clearly defining the meaning conveyed by natural languages. This paradox can be resolved by disambiguating the word “horse.” Does it refer to the “whole of all horse kind” (the mereological view) or to “horseness” (the Platonic view)? The mereological interpretation was famously—and controversially—introduced by Hansen (1983), see Fraser (2007) for a discussion of early Chinese ontological views of language.



Frontispiece of the OuCuiPian Library by Chevalier (1990). A different kind of cooking with letters.

Syrielle Montariol,\* Étienne Simon,\* Arij Riabi, Djamé Seddah. “Fine-tuning and Sampling Strategies for Multimodal Role Labeling of Entities under Class Imbalance” *CON-  
 STRAINT* 2022

\* Equal contributions

1135 pervised relation extraction. In Chapter 3, we propose a solution to train  
1136 deep relation extraction models in an unsupervised fashion. The problem  
1137 we tackle is a stability problem between a powerful universal approximator  
1138 and a weak supervision signal transpiring through the repetitions in the  
1139 data. This chapter was the object of a publication at ACL (Simon et al.  
1141 2019). Chapter 4 explores the methods to exploit the structure of the data  
1142 more directly through the use of graph-based models. In particular, we  
1143 draw parallels with the Weisfeiler–Leman isomorphism test to design new  
1144 methods using topological (dataset-level) and linguistic (sentence-level)  
1145 features jointly. Appendix A contains the state-mandated thesis summary  
1146 in French. The other appendices provide valuable information that can  
1147 be used as references. We strongly encourage the reader to refer to them  
1148 for additional details on the datasets (Appendix C), but even more so for  
1149 the list of assumptions made by relation extraction models (Appendix B).  
1151 These modeling hypotheses are central to the design of unsupervised ap-  
1152 proaches. In addition to their definition and reference to the introduc-  
1153 ing section, Appendix B provides counterexamples, which might help the  
1154 reader understand the nature of these assumptions.  
1156

Étienne Simon, Vincent Guigue, Benjamin Piwowarski. “Unsupervised Information Extraction: Regularizing Discriminative Approaches with Relation Distribution Losses” ACL 2019  
The work presented in Chapter 4 still needs to be polished with more experimental work and is yet unpublished at the time of writing.

1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188



## Chapter 1

# Context: Distributed Representations

Language conveys meaning. Thus, it should be possible to explicitly map a text to its semantic content. The research reported in this thesis seeks to algorithmically extract meaning conveyed by language using deep learning techniques from the information extraction and natural language processing (NLP) fields. We focus on the task of relation extraction, in which we seek to extract the semantic relation conveyed by a sentence. For example, given the sentence “Paris is the capital of France,” we seek to extract the relation “*capital of*.” To build a formal representation of relations, we use knowledge bases. In their simplest form, knowledge bases encode knowledge as a set of facts, which take the form (entity, relation, entity) such as (Paris, *capital of*, France). Like natural languages, knowledge bases purpose to convey meaning<sup>9</sup> but in a structure that is readily manipulable by algorithms. However, most knowledge—like this thesis—comes in the form of text. There lies the usefulness of the relation extraction task on which we focus. By “translating” natural language into knowledge bases, we seek to make more knowledge available to algorithms.

In this chapter, we focus on the two kinds of data we deal with in this thesis, namely text and knowledge bases. Subsequent chapters will deal with the extraction of knowledge base facts from text. In Section 1.1, we begin by positioning this task within the larger historical context by focusing on how the fields of machine learning, NLP and information extraction developed. Before delving into the specific algorithms for relation extraction, we must first define how to process language and how to represent semantic information in a way that can be manipulated by machine learning algorithms. In particular, we seek to obtain a *distributed representation*—which we define in the next section—of both language and knowledge bases since deep learning algorithms cannot directly work with non-distributed representations. We first inspect the representation of words in Section 1.2 before exploring how to process whole sentences in Section 1.3. Finally, Section 1.4 focuses on knowledge bases by first giving a formal definition before studying methods for extracting distributed representations from them.

## 1.1 Historical Development

In this section, we expose the rationale for applying deep learning to relation extraction, how the related fields appeared and why the task is

“*Meaning is what essence becomes when it is divorced from the object of reference and wedded to the word.*”

— Willard Van Orman Quine, “Main Trends in Recent Philosophy: Two Dogmas of Empiricism” (1951)

Quine was skeptical that facts about the meanings of linguistic expressions existed, for a critical response to his position see Soames (1997).

“*In scientific discourse what matters are the solid facts of a matter, not elegance.*”

— Wang Chong, “Lunheng” Chapter 85 (circa. 80)

Adapted from the translation of Harbsmeier (1989), Chong promotes truth over elegance despite the influence of early Chinese skepticism.

<sup>9</sup> Knowledge bases usually focus on knowledge which can be seen as a subset of all possible meanings. For example, facts like (I, *want*, ice cream) are not usually encoded in knowledge bases. However, they theoretically could. To be precise, throughout this thesis we’ll be using knowledge bases in two ways:

- as a basic theoretical structured representation of meaning,
- as a practical datasets to evaluate algorithms on.

This means that algorithms tested on existing knowledge bases are only tested on a subset of possible meanings. However, when we discuss the representation of knowledge base facts, note that this can be generalized to any meaningful facts expressible in the knowledge base framework.

「論貴是而不務華」

1243 relevant. Since algorithms were first given to train generic deep neural net-  
 1244 works (Glorot et al. 2011; Geoffrey E. Hinton et al. 2006), most problems  
 1245 tackled by machine learning can now be approached with deep learning  
 1246 methods. Over the last few years, deep learning has been very success-  
 1247 ful in a variety of tasks such as image classification (Krizhevsky et al.  
 1248 2012), machine translation (Cho et al. 2014), audio synthesis (van den  
 1249 Oord et al. 2016), etc. This is why it is not surprising that deep learning is  
 1250 now applied to more tasks traditionally tackled by other machine learning  
 1251 methods, such as in this thesis, where we apply it to relation extraction.

1252 From a historical point of view, machine learning—and hence deep  
 1253 learning—are deeply anchored in *empiricism*. Empiricism is the epistemo-  
 1254 logical paradigm in which knowledge is anchored in sensory experiences  
 1255 of the world, which are called empirical evidence. This is not to say that  
 1256 there are no theoretical arguments motivating the use of certain machine  
 1257 learning methods; the universal approximation theorems (Cybenko 1989;  
 1258 Leshno et al. 1993) can be seen as a theoretical argument for deep learn-  
 1259 ing. But in the end, a machine learning method draws its legitimacy from  
 1260 the observation that they perform strongly on a real dataset. This is in  
 1261 stark contrast to the rationalist paradigm, which posits that knowledge  
 1262 comes primarily from reason.

1263 This strong leaning on empiricism can also be seen in NLP. NLP comes  
 1264 from the *externalist* approach to linguistic theorizing, focusing its anal-  
 1265 yses on actual utterances. A linguistic tool that externalists often avoid  
 1266 while being widely used by other schools is elicitation through prospective  
 1267 questioning: “Is this sentence grammatical?” Externalists consider that  
 1268 language is acquired through distributional properties of words and other  
 1269 constituents;<sup>10</sup> and study these properties by collecting corpora of nat-  
 1270 urally occurring utterances. The associated school of structural linguistics  
 1271 inscribes itself into the broader view of *structuralism*, the belief that  
 1272 phenomena are intelligible through a concept of structure that connects  
 1273 them together, the focus being more on these interrelations instead of  
 1274 each individual object. In the case of linguistics, this view was pioneered  
 1275 by Ferdinand de Saussure which stated in its course in general linguistics:

1276 Language is a system whose parts can and must all be consid-  
 1277 ered in their synchronic<sup>11</sup> solidarity.

1278 — Ferdinand de Saussure, *Cours de*  
 1279 *linguistique générale* (1916)

1280 This train of thought gave rise to *distributionalism* whose ideas are best  
 1281 illustrated by the distributional hypothesis stated in Harris (1954):

1283 **Distributional Hypothesis:** *Words that occur in similar contexts convey*  
 1284 *similar meanings.*

1285 This can be pushed further by stating that a word is solely characterized  
 1286 by the context in which it appears.

1287 On the artificial intelligence side, deep learning is usually compared  
 1288 to symbolic approaches. The distinction originates in the way information  
 1289 is represented by the system. In the symbolic approach, information is  
 1290 carried by strongly structured representations in which a concept is usu-  
 1291 ally associated with a single entity, such as a variable in a formula or in  
 1292 a probabilistic graphical model. On the other hand, deep learning uses  
 1293 distributed representations in which there is a many-to-many relationship  
 1294 between concepts and neurons; each concept is represented by many neu-  
 1295 rons, and each neuron represents many concepts. The idea that mental  
 1296

<sup>10</sup> In other words, language is acquired by observing empirical co-occurrences: where words go and where they don’t in actual utterances tell us where they can go and where they can’t.

“ La langue est un système dont toutes les parties peuvent et doivent être considérées dans leur solidarité synchronique.

— Ferdinand de Saussure, *Cours de linguistique générale* (1916)

<sup>11</sup> Saussure makes a distinction between synchronic—at a certain point in time—and diachronic—changing over time—analyses. This does not mean that the meaning of a word is not influenced by its history, but that this influence is entirely captured by the relations of the word with others at the present time and that conditioned on these relations, the current meaning of the word is independent of its past meaning.

1297 phenomena can be represented using this paradigm is known as *connec-*  
 1298 *tionism*. One particular argument in favor of connectionism is the ability  
 1299 to degrade gracefully: deleting a unit in a symbolic representation equates  
 1300 to deleting a concept, while deleting a unit in a distributed representation  
 1301 merely lowers the precision with which concepts are defined. Note that  
 1302 connectionism is not necessarily incompatible with a symbolic theory of  
 1303 cognition. Distributed representations can be seen as a low-level explanation  
 1304 of cognition, while from this point of view, symbolic representation is  
 1305 a high-level interpretation encoded by distributed representations.<sup>12</sup>

1306 Furthermore, we can make a distinction on how structured is the kind  
 1307 of data used. In this thesis, we will especially focus on the relationship  
 1308 between unstructured text<sup>13</sup> and structured data (in the form of knowledge  
 1309 bases). To give a sense of this difference, compare the following text from  
 1310 the Paris Wikipedia page to facts from the Wikidata knowledge base:

1311		
1312	Paris is the capital and most	Paris <i>capital of</i> France
1313	populous city of France. The	
1314	City of Paris is the centre and	Paris <i>located in the adminis-</i>
1315	seat of government of the region	<i>trative territorial entity</i> Île-de-
1316	and province of Île-de-France.	France
1317		

1318 Through this example, we see that both natural languages and knowl-  
 1319 edge bases encode meaning. To talk about what they encode, we assume  
 1320 the existence of a semantic space containing all possible meanings. We do  
 1321 not assume any theory of meaning used to define this space; this allows us  
 1322 to stay neutral on whether language is ontologically prior to propositional  
 1323 attitudes and its link with reality or semantically evaluable mental states.  
 1324 In the same way that different natural languages are different methods  
 1325 to address this semantic space, knowledge bases seek to refer to the same  
 1326 semantic space<sup>14</sup> with an extremely rigid grammar.

1327 Both natural language and knowledge bases are discrete systems. For  
 1328 both these systems, we can use the distributional hypothesis to obtain  
 1329 continuous distributed representations. These representations purpose to  
 1330 capture the semantic as a simple topological space such as a Euclidean  
 1331 vector space where distance encodes dissimilarity, as shown in Figure 1.1.  
 1332 Moreover, using a differentiable manifold allows us to train these repre-  
 1333 sentations through backpropagation using neural architectures.

1334 The question of how to process texts algorithmically has evolved over  
 1335 the last fifty years. Language being conveyed through symbolic representa-  
 1336 tions, it is quite natural for us to manipulate them. As such, early machine  
 1337 learning models strongly relied on them. For a long time, symbolic ap-  
 1338 proaches had an empirical advantage: they worked better. However, in the  
 1339 last few years, distributed representations have shown unyielding results,  
 1340 and most tasks are now tackled with deep learning using distributed rep-  
 1341 resentations. As an example, this can be seen in the machine translation  
 1342 task. Early models from the 1950s onward were rule-based. Starting in the  
 1343 1990s, statistical approaches were used, first using statistics of words then  
 1344 of phrases. Looking at the Workshop on statistical machine translation  
 1345 (WMT): at the beginning of the last decade, no neural approaches were used  
 1346 and the report (Callison-Burch et al. 2010) deplored the disappearance of  
 1347 rule-based systems, at the end of the decade, most systems were based on  
 1348 distributed representations (Barrault et al. 2020).<sup>15</sup> While this transition  
 1349 occurred in NLP, knowledge representation has been a stronghold of sym-  
 1350 bolic approaches until very recently. The research reported in this thesis

<sup>12</sup> This view on the relation between distributed and symbolic representa-  
 tions can be seen in the early neural networks literature as can be seen in  
 Geoffrey E Hinton (1986), which is often cited for its formalization of the  
 backpropagation algorithm. More recently, Greff et al. (2020) investigate  
 the binding problem between symbols and distributed representations.

<sup>13</sup> Of course, language does have a structure. We do not deny the existence  
 of grammar but merely state that text is less structured than other struc-  
 tures studied in this chapter (see Section 1.4).

We use *slanted text* to indicate a relational surface form such as “*capital of*”  
 in the fact “Paris *capital of* France.”

<sup>14</sup> Strictly speaking, practical knowledge bases only seek to index a subset  
 of this space, see note 9 in the margin of page 25.

This transition from rule-based models to statistical models to neural network  
 models can also be seen in relation extraction with Hearst (1992, symbolic  
 rule-based, Section 2.2.1), SIFT (1998, symbolic statistical, Section 2.3.4) and  
 PCNN (2015, distributed neural, Section 2.3.6).

<sup>15</sup> To be more precise, most models use transformers which are a kind of neural  
 network introduced in Section 1.3.4.

1351 aims to develop the distributed approach to knowledge representation for  
 1352 the task of relation extraction. In the remainder of this chapter, we first  
 1353 report the distributed approaches to NLP, which showcased state-of-the-  
 1354 art results for the last decade, before presenting a structured symbolic  
 1355 representation, knowledge bases, and some methods to obtain distributed  
 1356 representations from them.

1357

1358

## 1359 1.2 Distributed Representation of Words

1360

1361

1362 Natural language processing (NLP) deals with the automatic manipulation  
 1363 of natural language by algorithms. Nowadays, a large pan of NLP concerns  
 1364 itself with the question of how to obtain good distributed representations  
 1365 from textual inputs. What constitutes a good representation may vary,  
 1366 but it is usually measured by performance on a task of interest. Natural  
 1367 language inputs present themselves as tokens or sequences of tokens, usu-  
 1368 ally in the form of words stringed together into sentences. The goal is then  
 1369 to map these sequences of symbolic units to distributed representations.  
 1370 This section and the next present several methods designed to achieve this  
 1371 goal which have become ubiquitous in NLP research. We first describe how  
 1372 to obtain good representations of words—or of smaller semantic units in  
 1373 Section 1.2.3—before studying how to use these representations to process  
 1374 whole sentences in Section 1.3.

1375 Given a vocabulary, that is a set of words  $V = \{a, \text{aardvark}, \text{aback}, \dots\}$ ,  
 1376 our goal is to map each word  $w \in V$  to an embedding  $u_w \in \mathbb{R}^d$  where  $d$  is a  
 1377 hyperparameter. An example of an embedding space is given in Figure 1.1.  
 1378 One of the early methods to embed words like this is latent semantic anal-  
 1379 ysis (LSA, Dumais et al. 1988). Interestingly, LSA was popularized by the  
 1380 information retrieval field under the name latent semantic indexing (LSI).  
 1381 The basis of LSA is a document–term matrix indicating how many times a  
 1382 word appears in a document. A naive approach would be to take the rows  
 1383 of this matrix; we would obtain a vector representation of each word, the  
 1384 dimension  $d$  of these embeddings would be the number of documents. The  
 1385 similarity of two words is then evaluated by taking the cosine similarity of  
 1386 the associated vectors; in the simple case described above, this value would  
 1387 be high if the two words often appear together in the same documents and  
 1388 low otherwise. We can already see that this representation is distributed  
 1389 since each document makes up a small fraction of the representation of  
 1390 the words it contains. However, this approach is not practical, as either  $d$   
 1391 is too large, or the representations obtained tend to be noisy (when the  
 1392 number of documents is relatively small). So LSA goes one step further and  
 1393 builds a low-rank approximation of this matrix such that  $d$  can be cho-  
 1394 sen as small as we want. This basic idea of modeling word co-occurrences  
 1395 forms the basis behind most word embedding techniques.

1396 In this section, we focus on the representation of words, yet most NLP  
 1397 tasks need to process longer chunks of text; this will be the focus of Sec-  
 1398 tion 1.3. We center our overview of word representations on word2vec in  
 1399 Section 1.2.1. With the advent of deep learning, word2vec has been the  
 1400 most ubiquitous word embedding technique. Additionally, it introduced  
 1401 negative sampling, a technique that we make use of in Chapter 3. Sec-  
 1402 tion 1.2.2 introduces the notion of language model, which is central to  
 1403 several representation extraction techniques in NLP; we also present sev-  
 1404 eral alternatives to word2vec used before the transition to sentence-level

In contrast, a symbolic representation of words would simply map each word to an index  $V \rightarrow \{1, \dots, |V|\}$ .

Dumais et al., “Using latent semantic analysis to improve access to textual information” SIGCHI 1988

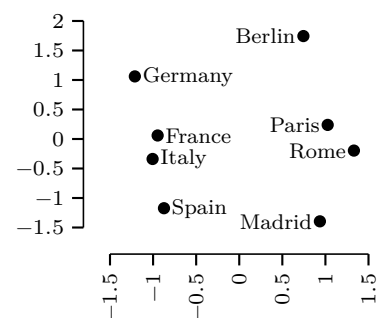


Figure 1.1: Selected word2vec embeddings of dimension  $d = 300$ , projected into two dimensions using PCA (explained variance ratio 27.6%+25.4%). The representations encode a strong separation between countries and capitals. Furthermore, the relative position of each country with respect to its associated capital is somewhat similar.

1405 approaches of Section 1.3.4. Finally, while models presented in this section  
 1406 are focused on words, smaller semantic units can similarly be used. This  
 1407 is especially needed for languages in which words have a complex inter-  
 1408 nal structure, but it can also be applied to English. Section 1.2.3 will ex-  
 1409 plore alternative levels at which we can apply methods from Sections 1.2.1  
 1410 and 1.2.2.

### 1411 1412 1.2.1 Word2vec 1413

1414 Word2vec (Mikolov et al. 2013a,b) is one of the first NLP models widely  
 1415 used for the representations it produces. As its name implies, word2vec  
 1416 outputs word representations; however, its general framework can be used  
 1417 on other kinds of tokens. Word2vec relies strongly on the distributional  
 1418 hypothesis: its goal is to model the context of a word to produce a rep-  
 1419 resentation of the word itself, a technique which was pioneered by Bengio  
 1420 et al. (2003). Several variants of the word2vec model exist, but for the  
 1421 sake of conciseness, this section focuses on the skip-gram with negative  
 1422 sampling (SGNS) approach.

#### 1423 1424 1.2.1.1 Skip-gram 1425

1426 Given a word, the idea behind skip-gram is to model its context.<sup>16</sup> The  
 1427 probability of a word  $c \in V$  to appear in the context of a word  $w \in V$  is  
 1428 modeled by the following softmax:

$$1429 \quad P(c | w) = \frac{\exp(\mathbf{u}_w^\top \mathbf{u}'_c)}{\sum_{c' \in V} \exp(\mathbf{u}_w^\top \mathbf{u}'_{c'})} \quad (1.1)$$

1430 where  $V$  is the vocabulary, and  $\mathbf{U}, \mathbf{U}' \in \mathbb{R}^{V \times d}$  are the model parameters  
 1431 assigning a vector representation to all words in the vocabulary. The rows  
 1432 of these parameters  $\mathbf{u}_w$  and  $\mathbf{u}'_w$  are what is of interest when word2vec  
 1433 is used for transfer learning. Once the model has been trained,  $\mathbf{u}_w$  can  
 1434 be used as a distributed representation for  $w$ , capturing its associated  
 1435 semantics. See Figure 1.1 for an example of extracted vectors.

#### 1436 1437 1438 1439 1440 1.2.1.2 Noise Contrastive Estimation

1441 Evaluating Equation 1.1 is quite expensive since the normalization term in-  
 1442 volves all the words in the vocabulary. Noise Contrastive Estimation (NCE,  
 1443 Gutmann and Hyvärinen 2010) is a training method that removes the need  
 1444 to compute the partition function of probabilistic models explicitly. To  
 1445 achieve this, NCE reframes the model as a binary classification problem by  
 1446 modeling the probability that a data point—in word2vec’s case a word-  
 1447 context pair—comes from the observed dataset  $P(D = 1 | w, c)$ . This prob-  
 1448 ability is contrasted with  $k$  samples from a noise distribution following the  
 1449 unigram distribution  $\hat{P}(W)$ , that is the empirical word frequency.<sup>17</sup> This  
 1450 translate to  $P(c | D = 1, w) = \hat{P}(c | w)$  and  $P(c | D = 0, w) = \hat{P}(W = c)$ .  
 1451 Using the prior  $P(D = 0) = \frac{k}{k+1}$ , the posterior can be expressed as:

$$1452 \quad P(D = 1 | w, c) = \frac{\hat{P}(c | w)}{\hat{P}(c | w) + k\hat{P}(c)}. \quad (1.2)$$

1453  
1454 Restating Equation 1.1 as  $P(c | w) = \exp(\mathbf{u}_w^\top \mathbf{u}'_c) \times \gamma_w$  and treating  $\gamma_w$   
 1455 as another model parameter, NCE allows us to train  $\mathbf{U}$  and  $\mathbf{U}'$  without

Mikolov et al., “Distributed Representations of Words and Phrases and their Compositionality” NeurIPS 2013

Bengio et al., “A Neural Probabilistic Language Model” JMLR 2003

<sup>16</sup> The context of a word  $w$  is defined as all words appearing in a fixed-size window around  $w$  in the text. In the case of word2vec, this window is of size five in both directions.

Here, we omit the conditioning on the parameters. More formally,  $P(c | w)$  should be written  $P(c | w; \mathbf{U}, \mathbf{U}')$ .

Gutmann and Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models” AISTATS 2010

We use  $\hat{P}$  to refer to empirical distributions, whereas  $P$  denotes a modeled probability. For example,  $\hat{P}(c | w)$  is the actual frequency of the word  $c \in V$  in the context of  $w \in V$ . While  $P(c | w)$  is the probability word2vec assigns to a given pair  $(c, w) \in V^2$ .

<sup>17</sup> Word2vec actually scales this distribution and uses various other tricks to lessen the effect of frequent words, refer to Mikolov et al. (2013b) for details.

1459 computing the denominator of Equation 1.1. Furthermore, estimating  $\gamma_w$   
 1460 is not even necessary, since Mnih and Teh (2012) showed that using  $\gamma_w = 1$   
 1461 for all  $w$  works well in practice. The final objective maximised by NCE is  
 1462 the log-likelihood of the classification data:

$$1463$$

$$1464 \quad J_{\text{NCE}}(w, c) = \log P(D = 1 \mid w, c) + \sum_{i=1}^k \mathbb{E}_{c'_i \sim P(W)} [\log P(D = 0 \mid w, c'_i)].$$

$$1465 \quad \quad \quad (1.3)$$

1467 Gutmann and Hyvärinen (2010) showed that optimizing  $J_{\text{NCE}}$  is equiv-  
 1468 alent to maximizing the log-likelihood using Equation 1.1 under some rea-  
 1469 sonable assumptions.

### 1471 1.2.1.3 Negative Sampling

1472 However, SGNS uses a different approximation of Equation 1.1 called neg-  
 1473 ative sampling. The difference is mainly visible in the expression of the  
 1474 objective which simplifies to:  
 1475

$$1476$$

$$1477 \quad J_{\text{NEG}}(w, c) = \log \sigma(\mathbf{u}_w^\top \mathbf{u}'_c) + \sum_{i=1}^k \mathbb{E}_{c'_i \sim P(W)} [\log \sigma(-\mathbf{u}_w^\top \mathbf{u}'_{c'_i})].$$

$$1478 \quad \quad \quad (1.4)$$

1480 This can be shown to be similar to NCE, where Equation 1.2 is instead  
 1481 replaced by the following posterior:

$$1482$$

$$1483 \quad P(D = 1 \mid w, c) = \frac{\hat{P}(c \mid w)}{\hat{P}(c \mid w) + 1}.$$

$$1484 \quad \quad \quad (1.5)$$

1485  
 1486  
 1487 Optimizing the objective of Equation 1.4 is not equivalent to maxi-  
 1488 mizing the log-likelihood of the language model. But even though this is  
 1489 not an approximation of the softmax of Equation 1.1, this method has  
 1490 proven to be quite effective at producing good word representations. Levy  
 1491 and Goldberg (2014) explain the effectiveness of word2vec by showing  
 1492 that SGNS can be interpreted as factoring the pointwise mutual informa-  
 1493 tion (PMI) matrix between words and contexts. This led to the emergence  
 1494 of GloVe (Pennington et al. 2014), which produces word embeddings by  
 1495 directly factorizing the PMI matrix.

1496 The negative sampling algorithm is one of the main contributions of  
 1497 word2vec; it can be used outside NLP to optimize softmax over large do-  
 1498 mains. In particular, we make use of negative sampling to approximate a  
 1499 softmax over a large number of entities in Chapter 3. Furthermore, even  
 1500 though it was initially presented on words, the algorithm can be used on  
 1501 other kinds of tokens, as we will see in Section 1.2.3.

1502  
 1503

## 1504 1.2.2 Language Modeling for Word Representation

1505  
 1506 Word2vec is part of a large class of algorithms that seek to learn word  
 1507 representation from raw text. More precisely, to obtain distributed rep-  
 1508 resentations of natural language inputs, most modern approaches rely on  
 1509 language models. A language model specifies a probability distribution  
 1510 over sequences of tokens  $P(w_1, \dots, w_m)$ . The tokens  $w$  are usually words,  
 1511 but as we see in Section 1.2.3, they need not be. This distribution is of-  
 1512 ten decomposed into a product of conditional distributions on tokens. The

Levy and Goldberg, “Neural Word Embedding as Implicit Matrix Factorization” NeurIPS 2014

1513 most common approach is the so-called *causal* language model, which uses  
 1514 the following decomposition:

$$1515 \quad P(w_1, \dots, w_m) = \prod_{t=1}^m P(w_t \mid w_1, \dots, w_{t-1}). \quad (1.6)$$

1519 Modeling the tokens one by one cannot only enable the model to factorize  
 1520 the handling of local information but also makes it easy to sample to gener-  
 1521 ate new utterances. However most language models do not use an exact  
 1522 decomposition but either approximate  $P(\mathbf{w})$  directly or use the decompo-  
 1523 sition of Equation 1.6 together with an approximation of the conditionals  
 1524  $P(w_t \mid w_1, \dots, w_{t-1})$ . This is for example the case of word2vec which condi-  
 1525 tions each word on its close neighbors instead of using the whole sentence.

1526 The use of language models is motivated by transfer learning, the idea  
 1527 that by solving a problem, we can get knowledge about a different but  
 1528 related problem. To assign a probability to a sequence, language models  
 1529 extract intermediate latent factors, which were proven to capture the se-  
 1530 mantic information contained in the sequence. Using these latent factors  
 1531 as distributed representations for natural language inputs improved the  
 1532 performance of most NLP tasks. The effectiveness of language models can  
 1533 be justified by the externalist approach and the distributional hypothesis  
 1534 exposed in Section 1.1: a word is defined by the distribution of the other  
 1535 words with which it co-occurs.

1536 Since language models process sequences of words, we will delve into  
 1537 the details of these approaches in Section 1.3. Apart from the neural prob-  
 1538 abilistic language model of Bengio et al. (2003), a precursor to word em-  
 1539 bedding techniques was the CNN-based approach of Collobert and Weston  
 1540 (2008), both of them learn distributed word representations by approxi-  
 1541 mating  $P(\mathbf{w})$  using a window somewhat similar to word2vec.

1542 All of these methods learn *static* word embeddings, meaning that the  
 1543 vector assigned to a word such as “bank” is the same regardless of the  
 1544 context in which the word appears. In the last few years, *contextualized*  
 1545 word embeddings have grown in popularity; in these approaches, the word  
 1546 “bank” is assigned different embeddings in the phrases “robbing a bank”  
 1547 and “bank of a river.” These methods were first based on recurrent neural  
 1548 networks (Section 1.3.2) such as ELMO but are now primarily based  
 1549 on transformers (Section 1.3.4). Among contextualized word embedding  
 1550 built using transformers, some are based on the causal decomposition  
 1551 of Equation 1.6 (e.g. GPT) while others are based on masked language  
 1552 models (e.g. BERT), a different approximation of  $P(\mathbf{w})$  introduced in Sec-  
 1553 tion 1.3.4.2.

### 1554 1555 1.2.3 Subword Tokens

1557 We defined word2vec and language models for a vocabulary  $V$  composed of  
 1558 words. This may seem natural in the case of English and other somewhat  
 1559 analytic languages,<sup>18</sup> but it cannot directly be applied to all languages.  
 1560 Furthermore, language models that work at the word level tend to have  
 1561 difficulties working with rare words. A first solution to this problem is to  
 1562 use character-level models, but these tend to have a hard time dealing  
 1563 with the resulting long sequences.

1565 Modern approaches neither work at the word-level nor at the character-  
 1566 level; instead, an intermediate subword vocabulary is used. The standard

<sup>18</sup> An analytic language is a language with a low ratio of morphemes to words. This is in contrast to synthetic languages, where words have a complex inner structure. Take for example the Nahuatl word “Nimitztētla-maquiltīz” (I-you-someone-something-give-CAUSATIVE-FUTURE) meaning “I shall make somebody give something to you” (Suárez 1983). For this kind of language, word-level approaches fail. Older models preprocessed the text with a morphological segmentation algorithm, while modern approaches directly work on subword units.

method to build this vocabulary nowadays is to use the byte pair encoding algorithm (BPE, Gage 1994). BPE listed as Algorithm 1.1 consists in iteratively replacing the most common bigram  $c_1c_2$  in a corpus with a new token  $c_{\text{new}}$ . This new token can then appear in the most common bigram with another token  $c_{\text{new}}c_3$ , they are then replaced with a new token  $c'_{\text{new}}$  which represents a tri-gram in the original alphabet:  $c_1c_2c_3$ . This is repeated until the desired vocabulary size is reached. In this way, BPE extracts tokens close to morphemes, the smallest linguistic unit with a meaning. As an example, by using this algorithm, the word “pretrained” can be split into three parts: “pre-,” “-train-” and “-ed.”

Word2vec can be both applied to words and to subwords extracted by BPE or other algorithms. This is the case of fastText (Bojanowski et al. 2017) which uses the word2vec algorithm on fixed-size subwords. All the models discussed in this section and the next have very loose requirements on the vocabulary  $V$ . However, they might work best using a smaller  $V$ ; this is especially the case of transformers, the current state-of-the-art approach introduced in Section 1.3.4.

### 1.3 Distributed Representation of Sentences

Most NLP tasks are tackled at the sentence level. In the previous section, we saw how to obtain representations of words. We now focus on how to aggregate these word representations in order to process whole sentences. Henceforth, given a sentence of length  $m$ , we assume symbolic words  $\mathbf{w} \in V^m$  are embedded as  $\mathbf{X} \in \mathbb{R}^{m \times d}$  in a vector space of dimension  $d$ . This can be achieved through the use of an embedding matrix  $\mathbf{U} \in \mathbb{R}^{V \times d}$  such as the one provided by word2vec.

An early approach to sentence representation was to use *bag-of-words*, that is to simply ignore the ordering of the words. In this section, we focus on more modern, deep learning approaches. Section 1.3.1 presents CNNs, which process fixed-length sequences of words to produce representations of sentences. We then focus on RNNs in Section 1.3.2, a method to get representations of sentences through a causal language model. RNNs can be improved by an attention mechanism as explained in Section 1.3.3. Finally, we present transformers in Section 1.3.4, which build upon the concept of attention to extract state-of-the-art contextualized word representations.

#### 1.3.1 Convolutional Neural Network

Convolutional neural networks (CNN) can be used to build the representation of a sentence from the representation of its constituting words (Collobert and Weston 2008; Kim 2014). These words embeddings can come from word2vec (Section 1.2.1) or can be learned using a CNN with a language model objective (Section 1.2.2), the latter being the original approach proposed by Collobert and Weston (2008).

The basic idea behind CNN is to recognize patterns in a position-invariant fashion (Waibel et al. 1989). This is applicable to natural language following the principle of compositionality: the words composing an expression and the rules used to combine them determine its meaning, with little influence from the location of the expression in the text. So, given a sequence of  $d$ -dimensional embeddings  $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^d$ , a one

#### algorithm BPE

```

Inputs:  $n$  the vocabulary size
            $\mathbf{t}$  the corpus
Output:  $V$  the vocabulary

 $V \leftarrow$  all unique characters in  $\mathbf{t}$ 
while  $|V| < n$  do
     $c_1c_2 \leftarrow$  most common bigram
                    in  $\mathbf{t}$ 
     $c_{\text{new}} \leftarrow$  new token not in  $V$ 
     $\mathbf{t} \leftarrow$  replace all occurrences of
                 $c_1c_2$  in  $\mathbf{t}$  by  $c_{\text{new}}$ 
     $V \leftarrow V \cup \{c_{\text{new}}\}$ 
output  $V$ 

```

Algorithm 1.1: The byte pair encoding algorithm.

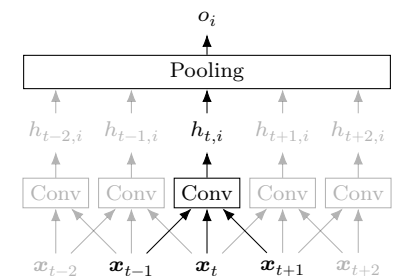


Figure 1.2: Architecture of a single convolutional filter with a pooling layer. The filter is of width 3, which means it works on trigrams. A single filter (the  $i$ -th) is shown here, this is repeated  $d'$  times, meaning that  $\mathbf{h}_t, \mathbf{o} \in \mathbb{R}^{d'}$ .

Collobert and Weston, “A unified architecture for natural language processing: deep neural networks with multitask learning” ICML 2008



1621 dimensional CNN works on the  $n$ -grams of the sequence, that is the sub-  
 1622 words<sup>19</sup>  $\mathbf{x}_{t:t+n-1} = (\mathbf{x}_t, \dots, \mathbf{x}_{t+n-1})$  of length  $n$ . The basic design of a CNN  
 1623 is illustrated in Figure 1.2. A convolutional layer is parametrized by  $d'$   
 1624 filters  $\mathbf{W}^{(i)} \in \mathbb{R}^{n \times d}$  of width  $n$  and a bias  $b^{(i)} \in \mathbb{R}$ . The  $t$ -th output of the  
 1625  $i$ -th filter layer is defined as:

$$1626 \quad h_t^{(i)} = f(\mathbf{W}^{(i)} * \mathbf{x}_{t:t+n-1} + b^{(i)}) \quad (1.7)$$

1628 where  $*$  is the convolution operator<sup>20</sup> and  $f$  is a non-linear function. As is  
 1629 usual with neural networks, several layers of this kind can be stacked. To  
 1630 obtain a fixed-size representation—which does not depend on the length of  
 1631 the sequence  $m$ —a pooling layer can be used. Most commonly, max-over-  
 1632 time pooling (Yamaguchi et al. 1990), which simply takes the maximum ac-  
 1633 tivation over time—that is sequence length—for each feature  $i = 1, \dots, d'$ .

1635 In the same way that word2vec produces a real vector space where  
 1636 words with similar meanings are close to each other, the sentence repre-  
 1637 sentations  $\mathbf{o}$  extracted by a CNN tend to be close to each other when the  
 1638 sentences convey similar meanings. This is somewhat dependent on the  
 1639 task on which the CNN is trained. However, the purpose of CNN is usually  
 1640 to extract the semantics of a sentence, and the nature of most tasks makes  
 1641 it so that sentences with similar meanings should have similar representa-  
 1642 tions.

### 1644 1.3.2 Recurrent Neural Network

1646 A limitation of CNNs is the difficulty they have modeling patterns of non-  
 1647 adjacent words. A second approach to process whole sentences is to use  
 1648 recurrent neural networks (RNN). RNNs purpose to sum up an entire sen-  
 1649 tence prefix into a fixed-size hidden state, updating this hidden state as  
 1650 the sentence is processed. This can be used to build a causal language  
 1651 model following the decomposition of Equation 1.6. As showcased by Fig-  
 1652 ure 1.3, the hidden state  $\mathbf{h}_t$  can be used to predict the next word  $w_{t+1}$   
 1653 with a simple linear layer followed by a softmax, formally:

$$1654 \quad \mathbf{h}_t = f(\mathbf{W}^{(x)} \mathbf{x}_t + \mathbf{W}^{(h)} \mathbf{h}_{t-1} + \mathbf{b}^{(h)}) \quad (1.8)$$

$$1655 \quad \hat{w}_t = \text{softmax}(\mathbf{W}^{(o)} \mathbf{h}_t + \mathbf{b}^{(o)})$$

1658 where  $\mathbf{W}^{(x)}$ ,  $\mathbf{W}^{(h)}$ ,  $\mathbf{W}^{(o)}$ ,  $\mathbf{b}^{(h)}$  and  $\mathbf{b}^{(o)}$  are model parameters and  $f$  is  
 1659 a non-linearity, usually a sigmoid  $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$ . This model is  
 1660 usually trained by minimizing the negative log-likelihood:

$$1662 \quad \mathcal{L}_{\text{RNN}}(\boldsymbol{\theta}) = \sum_{t=1}^m -\log P(w_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1}; \boldsymbol{\theta})$$

1665 using the backpropagation-through time algorithm. The gradient is run  
 1666 through all the steps of the RNN until reaching the beginning of the se-  
 1667 quence. When the sequence is a sentence, this can easily be achieved.  
 1668 However, when longer spans of text are considered, the gradient only goes  
 1669 back a fixed number of tokens in order to limit memory usage.

#### 1671 1.3.2.1 Long Short-term Memory

1673 Standard RNNs tend to have a hard time dealing with long sequences.  
 1674 This problem is linked to the vanishing and exploding gradient problems.

<sup>19</sup> Here we use *subwords* in its formal language theory meaning. In the simple setting where we deal with words in a sentence, this *subword* actually designates a sequence of consecutive words.

<sup>20</sup> Usually, a cross-correlation operator is actually used, which is equivalent up to a mirroring of the filters when they are real-valued.

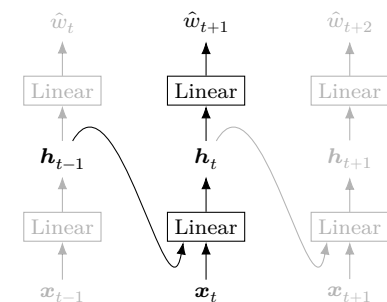
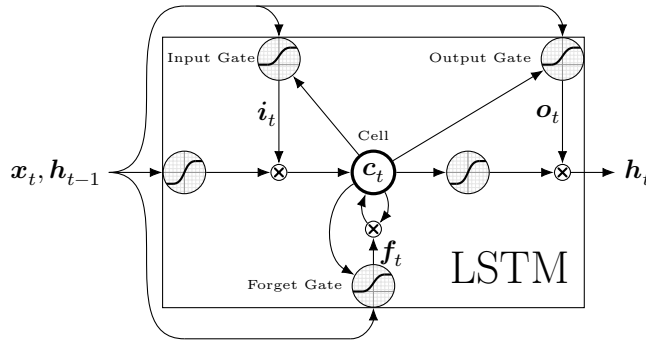


Figure 1.3: RNN language model unrolled through time.

We generally use  $\boldsymbol{\theta}$  to refer to the set of model parameters. In this case  $\boldsymbol{\theta} = \{\mathbf{W}^{(x)}, \mathbf{W}^{(h)}, \mathbf{W}^{(o)}, \mathbf{b}^{(h)}, \mathbf{b}^{(o)}\}$ .

1675 When the gradient goes through several non-linearities, it tends to be less  
 1676 meaningful, and gradient descent does not lead to satisfying parameters  
 1677 anymore. In particular, when  $\mathbf{W}^{(h)}$  has a large spectral norm, the values  
 1678  $\mathbf{h}_t$  tend to get bigger and bigger with long sequences, on the other hand  
 1679 when its spectral norm is small, these values get smaller and smaller. When  
 1680  $\mathbf{h}_t$  has a large magnitude, the sigmoid activation saturates and  $\frac{\partial \mathcal{L}_{\text{RNN}}}{\partial \mathbf{h}_t}$  gets  
 1681 close to zero, the gradient vanishes. RNN variants are used to alleviate  
 1682 this vanishing gradient problem, the most common being long short-term  
 1683 memory (LSTM, Hochreiter and Schmidhuber 1997).  
 1684



Hochreiter and Schmidhuber, “Long Short-Term Memory” Neco 1997

Figure 1.4: Architecture of an LSTM cell. In its simplest form, this block replaces the linear layer at the bottom of Figure 1.3. The link between  $\mathbf{c}_t$  and  $\mathbf{c}_{t-1}$  is illustrated by a self-loop but could be seen as an additional input and output.

1697 LSTMs redefine the recurrence of RNNs (Equation 1.8) by adding multi-  
 1698 plicative gates as illustrated by Figure 1.4. It is governed by the following  
 1699 set of equations:  
 1700

$$\begin{aligned}
 \mathbf{x}'_t &= \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} && \text{Recurrent input} \\
 \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}^{(c)} \mathbf{x}'_t + \mathbf{b}^{(c)}) && \text{Cell candidate} \\
 \mathbf{i}_t &= \sigma(\mathbf{W}^{(i)} \mathbf{x}'_t + \mathbf{U}^{(i)} \mathbf{c}_{t-1} + \mathbf{b}^{(i)}) && \text{Input gate} \\
 \mathbf{f}_t &= \sigma(\mathbf{W}^{(f)} \mathbf{x}'_t + \mathbf{U}^{(f)} \mathbf{c}_{t-1} + \mathbf{b}^{(f)}) && \text{Forget gate} \\
 \mathbf{c}_t &= \mathbf{i}_t \odot \tilde{\mathbf{c}}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} && \text{New cell} \\
 \mathbf{o}_t &= \sigma(\mathbf{W}^{(o)} \mathbf{x}'_t + \mathbf{U}^{(o)} \mathbf{c}_t + \mathbf{b}^{(o)}) && \text{Output gate} \\
 \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) && \text{Hidden layer output}
 \end{aligned}$$

$\odot$  is the element-wise multiplication and  $\sigma$  the sigmoid function.

As with RNN,  $\theta = \{\mathbf{W}^{(c)}, \mathbf{W}^{(i)}, \mathbf{U}^{(i)}, \mathbf{W}^{(f)}, \mathbf{U}^{(f)}, \mathbf{W}^{(o)}, \mathbf{U}^{(o)}, \mathbf{b}^{(c)}, \mathbf{b}^{(i)}, \mathbf{b}^{(f)}, \mathbf{b}^{(o)}\}$  are model parameters.

1714 The main peculiarity of LSTM is the presence of multiple gates used as  
 1715 masks or mixing factors in the unit. LSTM units are interpreted as having  
 1716 an internal cell memory  $\mathbf{c}_t$  which is an additional (internal) state alongside  
 1717  $\mathbf{h}_t$  and is used as input of the cell alongside  $\mathbf{x}_t$  and  $\mathbf{h}_{t-1}$ . When computing  
 1718 its activation, we first compute a cell candidate  $\tilde{\mathbf{c}}_t$  which is the potential  
 1719 successor to  $\mathbf{c}_t$ . Then, the multiplicative gates come into play, the cell  $\mathbf{c}_t$   
 1720 is partially updated with a mix of  $\mathbf{c}_{t-1}$  and  $\tilde{\mathbf{c}}_t$  controlled by the input and  
 1721 forget gates  $\mathbf{i}_t$  and  $\mathbf{f}_t$ . Finally, the output of the unit is masked by the  
 1722 output gate  $\mathbf{o}_t$ .<sup>21</sup>

1723 It has been theorized (Hochreiter 1998) that the gates are what makes  
 1724 LSTMs so powerful. The multiplications allow the model to learn to control  
 1725 the flow of information in the unit, thus counteracting the vanishing gra-  
 1726 dient problem. The basic building block of multiplicative gates has been  
 1727 reused for other RNN cell designs such as gated recurrent unit (GRU, Cho et  
 1728 al. 2014). Furthermore, random cell designs using multiplicative gates can

<sup>21</sup> Note that the output gate  $\mathbf{o}_t$  has its value computed from the new cell value  $\mathbf{c}_t$  instead of  $\mathbf{c}_{t-1}$  in contrast to the expression of  $\mathbf{i}_t$  and  $\mathbf{f}_t$ .

1729 be shown to perform as well as LSTM (Greff et al. 2017). However, standard  
1730 practice is to always use LSTM or GRU for recurrent neural networks.

1731

1732

1733 **1.3.2.2 ELMO**

1734 Recurrent neural networks with LSTM cells were widely used for language  
1735 modeling, both at the character-level (Sutskever et al. 2011) and at the  
1736 word-level (Jozefowicz et al. 2016). The first language model to become  
1737 widely used for extracting contextual word embeddings was ELMO (Em-  
1738 beddings from Language Model, Peters et al. 2018) which uses several  
1739 LSTM layers.

1740 The peculiarity of the word embeddings extracted by ELMO is that they  
1741 are contextualized (see Section 1.2.2). Static word embeddings models like  
1742 word2vec (Section 1.2.1) map each word to a unique vector. However, this  
1743 fares poorly with polysemic words and homographs whose meaning de-  
1744 pends on the context in which they are used. Contextualized word embed-  
1745 dings provide an answer to this problem. Given a sentence, ELMO proposes  
1746 to use the hidden states  $\mathbf{h}_t$  as a representation of each constituting word  
1747  $w_t$ . These representations are hence a function of the whole sentence.<sup>22</sup>  
1748 Thus words are mapped to different vectors in different contexts.

1749

1750

1751 **1.3.3 Attention Mechanism**

1752

1753 To obtain a vector representation of a sentence from an RNN, two straight-  
1754 forward methods are to use the last hidden state  $\mathbf{h}_m$  or use a pooling  
1755 layer similar to the one used in CNN, such as max-over-time pooling. How-  
1756 ever, both of these approaches present shortcomings: the last hidden state  
1757 tends to encode little information about the beginning of the sentence,  
1758 while pooling is too indiscriminate and influenced by unimportant words.  
1759 Using an attention mechanism is a way to avoid these shortcomings. Fur-  
1760 thermore, an attention mechanism is parametrized by a *query* which allows  
1761 us to select the piece of information we want to extract from the sentence.

1762 The concept of attention first appeared in neural machine translation  
1763 (NMT) under the name “alignment” (Bahdanau et al. 2015) before becom-  
1764 ing ubiquitous in NLP. The same principle was also presented under the  
1765 name *memory network* (Sukhbaatar et al. 2015; Weston et al. 2015). It is  
1766 also the building block of transformers, which are presented next. With  
1767 this in mind, we use the vocabulary of memory networks to describe the  
1768 attention mechanism.

1769

1770

1771 **1.3.3.1 Attention as a Mechanism for RNN**

1772

1773 The principle of an attention layer on top of an RNN is illustrated by  
1774 Figure 1.5. The layer takes three inputs: a query  $\mathbf{q} \in \mathbb{R}^d$ , memory keys  
1775  $\mathbf{K} \in \mathbb{R}^{\ell \times d}$  and memory values  $\mathbf{V} \in \mathbb{R}^{\ell \times d'}$ . Originally, more often than  
1776 not,  $\mathbf{K} = \mathbf{V}$ . In the model of Figure 1.5, the memory corresponds to the  
1777 hidden states of the RNN, which was the most common architecture when  
1778 attention was introduced in 2014. First, attention weights are computed  
1779 from the query  $\mathbf{q}$  and keys  $\mathbf{K}$ , then these weights are used to compute the  
1780 output  $\mathbf{o} \in \mathbb{R}^{d'}$  as a convex combination of the values  $\mathbf{V}$ :

1781

1782

$$\mathbf{o} = \text{softmax}(\mathbf{K}\mathbf{q})\mathbf{V}. \quad (1.9)$$

Peters et al., “Deep Contextualized Word Representations” NAACL 2018

Before ELMO, McCann et al. (2017) already trained contextualized word representations using an NMT task.

<sup>22</sup> In order to encode both the left and right context of a word, ELMO uses bidirectional LSTM, meaning that each layer contains two LSTM, one running from left-to-right and one right-to-left.

Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translate” ICLR 2015

Where softmax is a smooth version of the argmax function. It can also be seen as a multi-dimensional sigmoid, defined as:

$$\text{softmax}(\mathbf{x})_i = \frac{\exp x_i}{\sum_j \exp x_j}$$

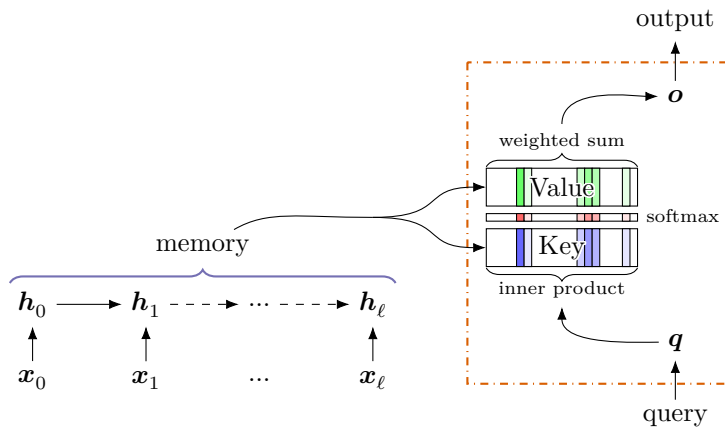


Figure 1.5: Schema of an attention mechanism. The attention scores are obtained by an inner product between the query and the memory. The output is obtained as a sum of the memory weighted by the softmax of the attention scores.

In NMT, the memory is built from the hidden states of an RNN running on the sentence to be translated (meaning  $\ell = m$ ), while the query is the state of the translated sentence (“what was already translated”), the attention is then recomputed for each output position. In other words, a new representation of the source sentence is recomputed for each word in the target sentence. The attention weights—that is, the output of the softmax—can provide an interpretation of what the model is focusing on when making a prediction. In the case of NMT, the attention for producing a translated word usually focuses on the corresponding word or group of words in the source sentence.

### 1.3.3.2 Attention as a Standalone Model

Since the attention mechanism produces a fixed-size representation ( $\mathbf{o}$ ) from a variable length sequence ( $\mathbf{K}, \mathbf{V}$ ), it can actually be used by itself without an RNN. This was already mentioned in Sukhbaatar et al. (2015) and used for language modeling. We now succinctly present their approach. As shown Figure 1.6, this is a causal language model (Section 1.2.2), at each step  $P(w_t | w_1, \dots, w_{t-1})$  is modeled. While the previous words constitute the memory of the attention mechanism, there is no natural value for the query. As such, for the first layer, it is simply taken to be a constant vector  $q_i^{(1)} = 0.1$  for all  $i = 1, \dots, d$ . When several attention layers are stacked, the output  $o^{(l)}$  of a layer  $l$  is used as the query  $q^{(l+1)}$  for the layer  $l + 1$ . Furthermore, residual connections with linear layers and modified ReLU non-linearities<sup>23</sup> are introduced between layers thus:  $q^{(l+1)} = \text{ReLU}_{\bullet}(\mathbf{W}^{(l)}q^{(l)} + o^{(l)})$  where the matrices  $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d}$  are parameters of the model. As usual, the next word prediction  $\hat{w}_i$  is made using a softmax layer.

**Temporal Encoding** The attention mechanism as described above is invariant to a permutation of the memory. This is not a problem when an RNN is run on the sentence, as it can encode the relative positions of each token. However, in the RNN-less approach of Sukhbaatar et al. (2015) this information is lost, which is quite damaging for language modeling. Indeed, this would mean that shuffling the words in a sentence—like inverting the subject and object of a verb—does not change its meaning. In order to solve this problem, temporal encoding is introduced. When predicting

Sukhbaatar et al., “End-To-End Memory Networks” NeurIPS 2015

<sup>23</sup> While the standard ReLU activation (Glorot et al. 2011) is defined as  $\text{ReLU}(x) = \max(0, x)$ . The non-linearity used in this model is  $\text{ReLU}_{\bullet}$ , which applies the ReLU activation to half of the units in the layer.

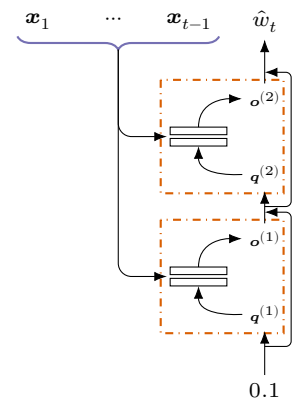


Figure 1.6: Schema of a memory network language model with two layers. Each red block corresponds to an attention mechanism as illustrated by Figure 1.5.

1837  $w_i$ , each word embedding  $\mathbf{x}_j$  in the memory is summed with a relative  
 1838 position embedding  $\mathbf{e}_{i-j}$ . These position embeddings are trained through  
 1839 back-propagation like any other parameters.

1840  
 1841 Attention mechanisms form the basis of current state-of-the-art ap-  
 1842 proaches in NLP. One of the explanations behind their success is that, in  
 1843 a sense, they are more shallow than RNN. Indeed, when computing  $\frac{\partial \hat{w}_i}{\partial \mathbf{x}_j}$   
 1844 for the language model of Sukhbaatar et al. (2015), one can see that part  
 1845 of the gradient goes through few non-linearities. In contrast, the infor-  
 1846 mation from  $\mathbf{x}_j$  to  $\hat{w}_i$  must go through the composition of at least  $i - j$   
 1847 non-linearities in an RNN, which may cause the gradient to vanish. How-  
 1848 ever, an attention mechanism has linear complexity in the length of the  
 1849 sequence for a total of  $\Theta(m \times d^2)$  operations at each step. When  $m$  is  
 1850 large, this can be prohibitive compared to RNN, which have a  $\Theta(d^2)$  com-  
 1851 plexity at each step. On the other hand, an attention layer can easily be  
 1852 parallelized while an RNN always necessitates  $\Omega(m)$  sequential operations.  
 1853

1854

### 1855 1.3.4 Transformers

1856  
 1857 Transformers (Vaswani et al. 2017) were originally introduced for NMT.  
 1858 Likewise to the memory network language model presented above, they  
 1859 introduce several slight modifications of its architecture which make them  
 1860 the current state of the art for most NLP tasks. For conciseness, we present  
 1861 the concept of transformers as used by BERT (Bidirectional Encoder Rep-  
 1862 resentations from Transformers, Devlin et al. 2019). BERT is a language  
 1863 model used to extract contextualized embeddings similar to ELMO but us-  
 1864 ing attention layers in place of LSTM layers.

Vaswani et al., “Attention is All you Need” NeurIPS 2017

Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” NAACL 2019

1865

#### 1866 1.3.4.1 Transformer Attention

1867  
 1868 The attention layers used by transformers are slightly modified. First, it is  
 1869 often advisable that in a neural network, all activations follow a standard  
 1870 normal distribution  $\mathcal{N}(0, 1)$ . In order to achieve this, transformers use  
 1871 scaled attention:

Note that in contrast to the classical attention mechanism presented in Section 1.3.3, transformers have  $\mathbf{K} \neq \mathbf{V}$ .

$$1872 \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left( \frac{\mathbf{K}\mathbf{q}}{\sqrt{d}} \right) \mathbf{V}. \quad (1.10)$$

1873  
 1874 This ensures that if  $\mathbf{K}$  and  $\mathbf{q}$  follow a standard normal distribution, so  
 1875 does the input of the softmax.  
 1876

1877 Second, multi-head attention is used: each layer actually applies  $h =$   
 1878 8 attentions in parallel. To ensure each individual attention captures a  
 1879 different part of the semantic, its input is projected by different matrices,  
 1880 one for each attention head:  
 1881

$$1882 \text{MultiHeadAttention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \begin{bmatrix} \text{head}_1(\mathbf{q}, \mathbf{K}, \mathbf{V}) \\ \text{head}_2(\mathbf{q}, \mathbf{K}, \mathbf{V}) \\ \vdots \\ \text{head}_h(\mathbf{q}, \mathbf{K}, \mathbf{V}) \end{bmatrix} \mathbf{W}^{(o)}$$

$$1883 \text{head}_i(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \text{Attention}(\mathbf{q}\mathbf{W}_i^{(q)}, \mathbf{K}\mathbf{W}_i^{(k)}, \mathbf{V}\mathbf{W}_i^{(v)}).$$

1884  
 1885  
 1886  
 1887  
 1888  
 1889 Lastly, on top of each attention layer is a linear layer with ReLU acti-  
 1890 vation and a linear layer followed by layer normalization (Ba et al. 2016).



## 1.4 Knowledge Base

Our goal is to extract structured knowledge from text. In this section, we introduce the object we use to express this knowledge, namely the knowledge base. A knowledge base is a symbolic semantic representation of some piece of knowledge. It is defined by a set of concepts, named *entities*, and by the relationships linking these entities together, named *facts* or *statements*. Formally, a knowledge base is constructed from a set of entities  $\mathcal{E}$ , a set of relations  $\mathcal{R}$  and a set of facts  $\mathcal{D}_{\text{KB}} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ . Note that these facts purpose to encode some kind of truth about the world. To illustrate, here are some examples from Wikidata (Vrandečić and Krötzsch 2014):

$$\begin{aligned} \mathcal{E} &= \{\text{Q90}(\text{Paris}), \text{Q7251}(\text{Alan Turing}), \dots\} \\ \mathcal{R} &= \{\text{P1376}(\text{capital of}), \text{P19}(\text{place of birth}), \dots\} \\ \mathcal{D}_{\text{KB}} &= \{\text{Q90 P1376 Q142} \text{ (Paris is the capital of France)}, \\ &\quad \text{Q3897 P1376 Q916} \text{ (Luanda is the capital of Angola)}, \\ &\quad \text{Q7251 P19 Q122744} \text{ (Alan Turing was born in Maida Vale)}, \\ &\quad \text{Q164047 P19 Q23311} \text{ (Alexander Pope was born in London)}, \\ &\quad \dots\} \end{aligned}$$

As indicated by the identifiers such as Q7251, knowledge bases link concepts together. An entity is a concept that may have several textual representations—surface forms—such as “Alan Turing” and “Alan Mathison Turing.” Here, we showed the Wikidata identifier whose purpose is to identify concepts uniquely. For ease of reading, when there is no ambiguity between an entity and one of its surface forms, we simply write the surface form without the identifier of its associated concept.

Given two entities  $e_1, e_2 \in \mathcal{E}$  and a relation  $r \in \mathcal{R}$ , we simply write  $e_1 r e_2$  as a shorthand notation for  $(e_1, r, e_2) \in \mathcal{D}_{\text{KB}}$ , meaning that  $r$  links  $e_1$  and  $e_2$  together. As illustrated by Figure 1.8,  $e_1$  is called the *head entity* of the fact or *subject* of the relation  $r$ . Similarly,  $e_2$  is called the *tail entity* or *object*, while  $r$  is called the *relation*, *property* or *predicate*.<sup>25</sup>

Thanks to this extremely rigid structure, knowledge bases are easier to process algorithmically. Querying some piece of information from a knowledge base is well defined and formalized. Query languages such as SPARQL ensure that information can be retrieved deterministically. This is in contrast to natural language, where querying some knowledge from a piece of text needs to be performed using an NLP model, thus incurring some form of variance on the result. With this in mind, it is not surprising that several machine learning models rely on knowledge bases to remove a source of uncertainty from their system; this can be done in a variety of tasks such as question answering (Berant et al. 2013; Yih et al. 2015), document retrieval (Dalton et al. 2014) and logical reasoning (Socher et al. 2013).

Commonly used general knowledge bases include Freebase (Bollacker et al. 2008), DBpedia (Auer et al. 2008) and Wikidata (Vrandečić and Krötzsch 2014). There are also several domain-specific knowledge bases such as Wordnet (G. A. Miller 1995) and GeneOntology (Gene Ontology Consortium 2004). Older works focus on Freebase—which is now discontinued—while newer ones focus on Wikidata and DBpedia. These knowledge bases usually include more information than what was described above. For example, Wikidata includes statement qualifiers that

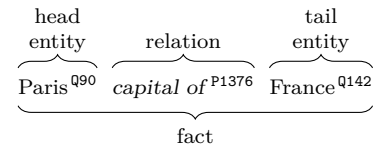


Figure 1.8: Structure of a knowledge base fact.

<sup>25</sup> The term *predicate* can either refer to the relation  $r$ , or to the couple  $(r, e_2)$ , thus we will avoid using this terminology.

Example of SPARQL query for all capital cities in Asia:

```
SELECT ?capital
WHERE {
  ?capital capital of ?country.
  ?country part of Asia.
}
```

1999 may modify a statement, such as the fact “Versailles capital of France”  
 2000 qualified by “end time: 5 October 1789.” For the sake of simplicity, we limit  
 2001 ourselves to triplets in  $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ . Further details on the specific knowledge  
 2002 bases can be found in Appendix C.

2003

2004

### 2005 1.4.1 Relation Algebra

2006 Relations linking two entities from the same set of entities  $\mathcal{E}$  are called  
 2007 binary endorelations. A relation such as “*capital of*” is a subset of the  
 2008 cartesian square  $\mathcal{E}^2$ ; it is a set of pairs of entities linked together by this  
 2009 relation. The set of all possible such relations exhibit a structure called a  
 2010 relation algebra  $(2^{\mathcal{E}^2}, \cap, \cup, \bar{\cdot}, \mathbf{0}, \mathbf{1}, \bullet, \mathbf{I}, \tilde{\cdot})$ . We use it as a formalized system  
 2011 of notation for relation properties. A relation algebra is defined from:

2012

2013

2014

2015

2016

2017

2018

2019

2020

2021

2022

2023

2024

2025

2026

2027

2028

2029

2030

2031

2032

2033

2034

2035

2036

2037

2038

2039

2040

2041

2042

2043

2044

2045

2046

2047

2048

2049

2050

2051

2052

- three special relations:
  - $\mathbf{0}$ , the empty relation linking no entities together ( $e_1 \mathbf{0} e_2$  is always false);
  - $\mathbf{1}$ , the complete relation linking all entities together ( $e_1 \mathbf{1} e_2$  is always true);
  - $\mathbf{I}$ , the identity relation linking all entities to themselves ( $e_1 \mathbf{I} e_2$  is true if and only if  $e_1 = e_2$ ).
- two unary operators:
  - the complementary relation  $\bar{r}$  which links together entities not linked by  $r$ ;
  - the converse  $\tilde{r}$  which reverses the direction of the relation such that  $e_1 \tilde{r} e_2$  holds if and only if  $e_2 r e_1$  holds.
- three binary operators (in order of lowest precedence, to highest precedence):
  - disjunction  $e_1 (r_1 \cup r_2) e_2$ , either  $r_1$  or  $r_2$  link  $e_1$  with  $e_2$ ;
  - conjunction  $e_1 (r_1 \cap r_2) e_2$ , both  $r_1$  and  $r_2$  link  $e_1$  with  $e_2$ ;
  - composition  $e_1 (r_1 \bullet r_2) e_2$ , there exist  $e_3 \in \mathcal{E}$  such that both  $e_1 r_1 e_3$  and  $e_3 r_2 e_2$  hold.

Thanks to this framework, we can express several properties on knowledge base relations since  $\mathcal{R} \subseteq 2^{\mathcal{E}^2}$ . For example, the *functional* property can be stated as  $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ . A relation  $r$  is functional when for all entities  $e_1$  there is at most one entity  $e_2$  such that  $e_1 r e_2$  holds. The relation “*born in*” is functional since all entities are either born at a single place or not born at all. Taking the above definition this means that for all cities  $c$  if we take all entities who were born in  $c$  ( $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ ) and then ( $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ ) look at where these entities were born ( $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ ), we must be back to  $c$  and only  $c$  ( $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ ) or no such  $c$  shall exist ( $\tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$ ). We need to take the disjunction with  $\mathbf{I}$  since some entities were not born anywhere, for example  $e (\tilde{r} \bullet r) e$  is false when  $r$  is “*born in*” and  $e$  is “Mount Everest.”

Other common properties of binary relations can be defined this way. One particular property of interest is the restriction of the domain and codomain of relations. A lot of relations can only apply to a specific type of entity, such as locations or people. To express these properties, we use the notation  $\mathbf{1}_X \subseteq \mathbf{1}$  with  $X \subseteq \mathcal{E}$  to refer to the complete relation restricted to entities in  $X$ :  $\mathbf{1}_X = \{(x_1, x_2) \mid x_1, x_2 \in X\}$ . This allows us to define

The concept of relation algebra was theorized as a structure for logical systems. Developed by several famous mathematicians such as Augustus De Morgan, Charles Peirce and Alfred Tarski, it can be used to express ZFC set theory. Here we only use relation algebra as a formal framework to express properties of binary relations.

Note that  $\bullet$  composes relations in the opposite order of the function composition  $\circ$ . Indeed while  $f \circ g$  means that  $g$  is applied first, then  $f$  is applied, “*mother*  $\bullet$  *born in*” means that “*mother*” is first applied to the entity, then “*born in*” is applied to the result.



2053 left-restriction (restriction of the domain) and right-restriction (restriction  
2054 of the co-domain). Relevant properties are given in Table 1.1.

2055 Some relation properties recurring in the literature are the cardinal-  
2056 ity constraints. They can be defined as combinations of the injective and  
2057 functional properties:

2058 **Many-to-Many** ( $N \rightarrow N$ ) the relation is neither injective nor functional.  
2059 Examples: “author of,” “language spoken,” “sibling of.”

2060 **Many-to-One** ( $N \rightarrow 1$ ) the relation is functional but it is not injective.  
2061 Examples: “place of birth,” “country.”

2062 **One-to-Many** ( $1 \rightarrow N$ ) the relation is injective but it is not functional.  
2063 Examples: “contains administrative territorial entity,” “has part.”

2064 **One-to-One** ( $1 \rightarrow 1$ ) the relation is both injective and functional.  
2065 Examples: “capital,” “largest city,” “highest point.”

2066 When a relation  $r$  is one-to-many, its converse  $\check{r}$  is many-to-one. The  
2067 usual way to design relations in knowledge bases is to use many-to-one  
2068 relations, making one-to-many relations quite rare in practice. Since most  
2069 systems handle relations in a symmetric fashion, this has little to no effect.

2070 Most of the examples given above are not strictly true. A person can  
2071 be both registered as being born in Paris and in France. Some countries do  
2072 not designate a single capital or share their highest point with a neighbor.  
2073 However, defining these properties is helpful to evaluate the abilities of  
2074 models to capture these kinds of relations. To handle such cases, these  
2075 properties can be seen in a probabilistic way.<sup>26</sup>

2076 We use the notations from relation algebra to formalize assumptions  
2077 made on the structure of knowledge bases. For example several models  
2078 assume that  $\forall r_1, r_2 \in \mathcal{R} : r_1 \cap r_2 = \mathbf{0}$ , that is all pairs of entities are  
2079 linked by at most one relation. A list of common assumptions is provided  
2080 in Appendix B, it should prove useful from the Chapter 2 onwards. For  
2081 readers unfamiliar with relation algebra notations, we provide detailed  
2082 explanation of complex formulae in the margins throughout this thesis.

## 2083 1.4.2 Distributed Representation through Knowledge 2084 Base Completion

2085 One problem with knowledge bases is that they are usually incomplete.  
2086 However, given some information about an entity, it is usually possible  
2087 to infer additional facts about this entity. This is called *knowledge base  
2088 completion*. Sometimes this inference is deterministic. For example, if two  
2089 entities have the same two parents, we can infer that they are siblings.  
2090 Quite often, this reasoning is probabilistic. For example, the head of state  
2091 of a country usually lives in this country’s capital; this probability can be  
2092 further increased by facts indicating that previous heads of state died in  
2093 the capital, etc.

2094 The task of knowledge base completion is essential for our work be-  
2095 cause of two reasons. First of all, it is the standard approach to obtain  
2096 a distributed representation of knowledge base objects. Second, the mod-  
2097 els used to tackle this task are often reused as part of relation extraction  
2098 systems; this is the case of all approaches presented in this section.

2099 We define two sub-tasks of knowledge base completion: *relation predic-  
2100 tion* and *entity prediction*.<sup>27</sup> In the relation prediction task, the goal is to

Property	Condition
Injective	$r \bullet \check{r} \cup \mathbf{I} = \mathbf{I}$
Functional	$\check{r} \bullet r \cup \mathbf{I} = \mathbf{I}$
Symmetric	$r = \check{r}$
Transitive	$r \bullet r \cup r = r$
Left-restriction	$r \bullet \check{r} \cup 1_X = 1_X$
Right-restriction	$\check{r} \bullet r \cup 1_X = 1_X$

Table 1.1: Some fundamental relation properties expressed as conditions in relation algebra.

<sup>26</sup> Given empirical data, the propensity of a relation to be many-to-one can be measured with a conditional entropy  $H(e_2 \mid e_1, r)$ . An entropy close to zero means the relation tends to be many-to-one.

<sup>27</sup> In the literature, both of these tasks can be called “link prediction” and “knowledge graph completion.”

2107 predict the relation between two entities ( $e_1 ? e_2$ ), while entity prediction  
 2108 focuses on predicting a missing entity in a triplet ( $e_1 r ?$  or  $? r e_2$ ). His-  
 2109 torically, this is performed using symbolic approaches. For example, this  
 2110 task can be tackled using an inference engine relying on a human expert  
 2111 inputting logical rules such as:

$$2112 \quad e_1 \text{ parent of } e_2 \wedge e_1 \text{ parent of } e_3 \wedge e_2 \neq e_3 \iff e_2 \text{ sibling of } e_3,$$

2114 or using the relation algebra notation introduced in Section 1.4.1:

$$2116 \quad \overline{\text{parent of}} \bullet \text{parent of} \cap \bar{\mathbf{I}} = \text{sibling of}.$$

2118 However, listing all possible logical implications is not feasible. As with  
 2119 NLP, to tackle this problem, another approach is to leverage distributed  
 2120 representations. Some good early results were obtained by RESCAL, which  
 2121 we present in Section 1.4.2.2. But the problem started to gather a lot of in-  
 2122 terest in the deep learning community with TransE (Section 1.4.2.3) which  
 2123 encodes relations as translation in the semantic space. This was followed  
 2124 by several other approaches that encoded relations as other kinds of ge-  
 2125 ometric transformations. All the models presented in this section assume  
 2126 that the entities are embedded in a latent semantic space  $\mathbb{R}^d$  with a matrix  
 2127  $\mathbf{U} \in \mathbb{R}^{\mathcal{E} \times d}$  where  $d$  is an hyperparameter.

#### 2129 1.4.2.1 Selectional Preferences

2131 Selectional preferences is a simple formalism that purposes to encode each  
 2132 relation with two linear maps assessing the predisposition of an entity to  
 2133 appear as the head or tail of a relation in a true fact. This can be done  
 2134 using an energy formalism, where the energy of a fact is defined as:

$$2136 \quad \psi_{\text{sp}}(e_1, r, e_2) = \mathbf{u}_{e_1}^\top \mathbf{a}_r + \mathbf{u}_{e_2}^\top \mathbf{b}_r \quad (1.12)$$

2138 with  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{\mathcal{R} \times d}$  two matrices encoding the preferences of each relation  
 2139 for certain entities. This energy function can then be used to define the  
 2140 probability that a fact holds using a softmax:

$$2142 \quad P(e_1, r, e_2) \propto \exp \psi_{\text{sp}}(e_1, r, e_2), \quad (1.13)$$

2144 this is sufficient for entity and relation predictions as we can usually com-  
 2145 pute the partition function over the set of all entities or relations. If this  
 2146 is not feasible, a technique such as NCE (Section 1.2.1.2) or negative sam-  
 2147 pling (Section 1.2.1.3) can be used to approximate Equation 1.13. Still,  
 2148 selectional preferences do not encode the interaction of the head and tail  
 2149 entities. As such it is quite weak for entity prediction, thus more expressive  
 2150 models are needed.

#### 2152 1.4.2.2 RESCAL

2153 RESCAL (Nickel et al. 2011) purposes to model relations by a bilinear form  
 2154  $\mathcal{E} \times \mathcal{E} \mapsto \mathbb{R}$  in the semantic space of entities. In other words, each relation  
 2155  $r \in \mathcal{R}$  is represented by a matrix  $\mathbf{C}_r \in \mathbb{R}^{d \times d}$  with the training algorithm  
 2156 seeking to enforce the following property:

$$2158 \quad \mathbf{u}_{e_1}^\top \mathbf{C}_r \mathbf{u}_{e_2} = \begin{cases} 1 & \text{if } e_1 r e_2 \text{ holds} \\ 0 & \text{otherwise.} \end{cases}$$

Relation prediction is quite similar to our task of interest: relation extraction. The main difference being that relation prediction is defined on knowledge bases, while relation extraction takes natural language inputs. This parallel is exploited by the model presented in Chapter 3.

$e_2 \overline{\text{parent of}} e_1$  means that  $e_1$  is a parent of  $e_2$ . Adding a composition to this,  $e_2 \overline{\text{parent of}} \bullet \text{parent of } e_3$  means that the aforementioned  $e_1$  has a child  $e_3$ . This child  $e_3$  could be the same as  $e_2$ , this is why we take the conjunction with the complement of the identity relation  $\cap \bar{\mathbf{I}}$ , thus obtaining the relation *sibling of*.

Nickel et al., "A Three-Way Model for Collective Learning on Multi-Relational Data" ICML 2011

2161 This can be seen as trying to factorize the tensor of facts  $\mathbf{X}$  as  $\mathbf{UCU}^\top$ ,  
 2162 where  $\mathbf{X} \in \{0, 1\}^{\mathcal{E} \times \mathcal{R} \times \mathcal{E}}$  with  $x_{e_1 r e_2} = 1$  if  $e_1 r e_2$  holds and  $x_{e_1 r e_2} = 0$   
 2163 otherwise. The parameters of the models  $\mathbf{U}$  and  $\mathbf{C}$  are trained using an al-  
 2164 ternating least-squares approach, minimizing a regularized reconstruction  
 2165 loss:

$$2166 \mathcal{L}_{\text{RESCAL}}(\mathbf{X}; \mathbf{U}, \mathbf{C}) = \frac{1}{2} \sum_{\substack{e_1, e_2 \in \mathcal{E} \\ r \in \mathcal{R}}} (x_{e_1 r e_2} - \mathbf{u}_{e_1}^\top \mathbf{C}_r \mathbf{u}_{e_2})^2 + \frac{1}{2} \lambda (\|\mathbf{U}\|_F^2 + \sum_{r \in \mathcal{R}} \|\mathbf{X}_r\|_F^2) \quad (1.14)$$

2170 Using bilinear forms allows RESCAL to capture entities interactions for  
 2171 each relation in a simple manner. However, the number of parameters to  
 2172 estimate grows quadratically with respect to the dimension of the semantic  
 2173 space  $d$ . This can be prohibitive as a large  $d$  is needed to ensure accurate  
 2174 modeling of the entities.

### 2176 1.4.2.3 TransE

2178 To find a balance between the number of parameters and the expressive-  
 2179 ness of the model, geometric approaches were developed starting with  
 2180 TransE (Bordes et al. 2013). TransE proposes to leverage the regularity  
 2181 exhibited by Figure 1.1 to embed both entities and relations in the same  
 2182 vector space. Formally, its assumption is that relations can be represented  
 2183 as translations between entities' embeddings. In addition to representing  
 2184 each entity  $e$  by an embedding  $\mathbf{u}_e \in \mathbb{R}^d$ , each relation  $r$  is also embedded  
 2185 as a translation in the same space as  $\mathbf{v}_r \in \mathbb{R}^d$ . The idea being that if  
 2186  $e_1 r e_2$  holds then  $\mathbf{u}_{e_1} + \mathbf{v}_r \approx \mathbf{u}_{e_2}$ . The authors argue that translations  
 2187 can represent hierarchical data by drawing a parallel with the embedding  
 2188 of a tree in an Euclidean plane—that is the usual representation of a tree  
 2189 as drawn on paper. As long as the distance between two levels in the tree  
 2190 is large enough, the children of a node are close together; this not only  
 2191 allows for the representation of one-to-many relations “child” but also for  
 2192 the many-to-many, symmetric and transitive relation “sibling” as the null  
 2193 translation.

2194 In order to enforce the translation property, a margin-based loss is  
 2195 used to train an energy-based model. The energy of true triplets drawn  
 2196 from the knowledge base is minimized, while negative triplets are sampled  
 2197 and have their energy maximized up to a certain margin. Given a positive  
 2198 triplet  $(e_1, r, e_2)$  and a negative triplet  $(e'_1, r, e'_2)$ , the TransE loss can be  
 2199 expressed as:

$$2201 \mathcal{L}_{\text{TE}}(e_1, r, e_2, e'_1, e'_2) = \max\left(0, \gamma + \Delta(\mathbf{u}_{e_1} + \mathbf{v}_r, \mathbf{u}_{e_2}) - \Delta(\mathbf{u}_{e'_1} + \mathbf{v}_r, \mathbf{u}_{e'_2})\right), \quad (1.15)$$

2203 where  $\Delta$  is a distance function such as the squared Euclidean distance  
 2204  $\Delta(\mathbf{u}_{e_1} + \mathbf{v}_r, \mathbf{u}_{e_2}) = \|\mathbf{u}_{e_1} + \mathbf{v}_r - \mathbf{u}_{e_2}\|_2^2$ . The negative triplets  $(e'_1, r, e'_2)$  are  
 2205 sampled by replacing one of the two entities of  $(e_1, r, e_2)$  by a random one  
 2206 which is sampled uniformly over all possible entities:

$$2208 N(e_1, e_2) = \begin{cases} (e_1, e') & \text{with probability 50\%} \\ (e', e_2) & \text{with probability 50\%} \end{cases}$$

2210 with  $e' \sim \mathcal{U}(\mathcal{E})$ .

2212 Since  $d$  is a distance, when the loss  $\mathcal{L}_{\text{TE}}$  is perfectly minimized, the  
 2213 positive part  $+\Delta(\mathbf{u}_{e_1} + \mathbf{v}_r, \mathbf{u}_{e_2})$  is 0. This means that the negative part

Bordes et al., “Translating Embed-  
 dings for Modeling Multi-relational  
 Data” NeurIPS 2013

2215  $-\Delta(\mathbf{u}_{e'_1} + \mathbf{v}_r, \mathbf{u}_{e'_2})$  contributes to the loss only when it is smaller than the  
 2216 margin  $\gamma$ . Since this criterion depends on the distance between entities,  
 2217 it can easily be optimized by increasing the entity embeddings norms. To  
 2218 avoid this degenerate solution, the entity embeddings are renormalized  
 2219 at each training step. The training loop and initialization procedure are  
 2220 detailed in Algorithm 1.2. Parameters  $\mathbf{U}$  and  $\mathbf{V}$  are optimized by stochastic  
 2221 gradient descent with early-stopping based on validation performance.

2222

2223

**Evaluation** The quality of the embeddings can be evaluated by measuring  
 2224 the accuracy of entity prediction based on them. Given a true triplet  
 2225  $(e_1, r, e_2) \in \mathcal{D}_{\text{KB}}$ , the energy  $\Delta(\mathbf{u}_{e'} + \mathbf{v}_r, \mathbf{u}_{e_2})$  is computed for all possible  
 2226 entities  $e' \in \mathcal{E}$ . The entity minimizing the energy is predicted as complet-  
 2227 ing the triplet. The same procedure is then applied on  $e_2$ . The correct  
 2228 entity minimizes the energy quite rarely, therefore in order to have a more  
 2229 informative score Bordes et al. (2013) reports the mean rank of the correct  
 2230 entity among all the entities ranked by the energy of their associated  
 2231 triplets. For reference, on WordNet, the mean rank of the correct entity is  
 2232 263 among 40 943 entities.

2233

2234

When expanding the expression  $\Delta(\mathbf{u}_{e_1} + \mathbf{v}_r, \mathbf{u}_{e_2})$  where  $d$  is the Eu-  
 2235 clidean distance, the main term ends up being  $\mathbf{u}_{e_1}^\top \mathbf{u}_{e_2} + \mathbf{v}_r^\top (\mathbf{u}_{e_2} - \mathbf{u}_{e_1})$ . As  
 2236 such, TransE captures all 2-way interactions between  $e_1$ ,  $r$  and  $e_2$ . How-  
 2237 ever, this means that 3-way interactions are not captured, this is how-  
 2238 ever standard in information extraction. Furthermore, TransE is unable  
 2239 to model several symmetric relations (when  $r = \tilde{r}$ ). To solve these prob-  
 2240 lems, other geometric transformations were proposed to improve TransE  
 2241 expressiveness, such as first projecting entities on a hyperplane (TransH,  
 2242 Z. Wang et al. 2014) or having the entities and relations live in different  
 2243 spaces (TransR, Y. Lin et al. 2015). Finally, all the methods mentioned  
 2244 in this section are not only useful for entity and relation predictions, but  
 2245 also as methods to obtain distributed representations of knowledge bases  
 2246 entities and relations. The matrices  $\mathbf{U}$  and  $\mathbf{V}$  learned by TransE can sub-  
 2247 sequently be used for other tasks involving knowledge bases, in the same  
 2248 way that transfer learning is used to obtain distributed representations of  
 2249 text using language models (Section 1.3.4.3).

2250

2251

2252

2253

2254

2255

2256

2257

2258

2259

2260

2261

2262

2263

2264

2265

2266

2267

2268

## 1.5 Conclusion

As exposed in Section 1.1, we are in the middle of a transition away from  
 symbolic representations towards distributed ones. We inscribe this thesis  
 within this transition. We deal with two kinds of symbolic representations  
 of meaning: unstructured language and structured knowledge bases. In  
 this chapter, we presented methods to extract distributed representations  
 for both of these systems. While in the following chapters, we will deal  
 with the link between language and knowledge bases.

Following word2vec (Section 1.2.1), feature extraction for textual inputs  
 is now mostly done through word embeddings. In order to obtain a  
 representation of a sentence, the models on top of these word embeddings  
 progressively evolved from CNN (Section 1.3.1) and RNN (Section 1.3.2) to-  
 wards transformers and contextualized word embeddings (Section 1.3.4).  
 As we will see in the next chapter, this trend was exactly followed by  
 relation extraction models.

### algorithm TRANS E

*Inputs:*  $\mathcal{D}_{\text{KB}}$  knowledge base  
 $\gamma$  margin  
 $d$  embedding dimension  
 $b$  batch size  
*Outputs:*  $\mathbf{U}$  entity embeddings  
 $\mathbf{V}$  relation embeddings

▷ *Initialization* ◁

$\mathbf{U} \leftarrow \mathcal{U}_{|\mathcal{E}| \times d} \left( -\frac{6}{\sqrt{d}}, \frac{6}{\sqrt{d}} \right)$

$\mathbf{V} \leftarrow \mathcal{U}_{|\mathcal{R}| \times d} \left( -\frac{6}{\sqrt{d}}, \frac{6}{\sqrt{d}} \right)$

$\forall r \in \mathcal{R} : \mathbf{v}_r \leftarrow \mathbf{v}_r / \|\mathbf{v}_r\|_2$

▷ *Training* ◁

**loop**

$\forall e \in \mathcal{E} : \mathbf{u}_e \leftarrow \mathbf{u}_e / \|\mathbf{u}_e\|_2$

$B \leftarrow \emptyset$

**for**  $i = 1, \dots, b$  **do**

    Sample  $(e_1, r, e_2) \sim \mathcal{U}(\mathcal{D}_{\text{KB}})$

    Sample  $(e'_1, e'_2) \sim N(e_1, e_2)$

$B \leftarrow B \cup \{(e_1, r, e_2, e'_1, e'_2)\}$

    Update  $\mathbf{U}$  and  $\mathbf{V}$  w.r.t.

$\nabla \sum_{(e_1, r, e_2, e'_1, e'_2) \in B} \mathcal{L}_{\text{TE}}(e_1, r, e_2, e'_1, e'_2)$

**output**  $\mathbf{U}, \mathbf{V}$

Algorithm 1.2: The TransE training algorithm. The relations are initialized randomly on the sphere but are free to drift away afterward, while entities are renormalized at each iteration. The loop updates parameters  $\mathbf{U}$  and  $\mathbf{V}$  using gradient descent and is stopped based on validation score. The gradient of  $\mathcal{L}_{\text{TE}}$  is computed from Equation 1.15.

2269 We then introduce the structured knowledge representation we handle  
2270 throughout this thesis, knowledge bases. In particular, Section 1.4.1 gives  
2271 a formal notation for handling relations which we use to write modeling  
2272 hypotheses in subsequent chapters. Finally, Section 1.4.2 presents common  
2273 models making use of distributed representations of knowledge bases for  
2274 the task of knowledge base completion. This task is not only the usual  
2275 evaluation framework for distributed knowledge base representations but  
2276 is also of special interest for Chapter 3, where we leverage the similarity  
2277 between the knowledge base completion and the relation extraction tasks.

2278 The progression of models presented in this chapter also reflects a  
2279 progression of the scale of problems. We started by exploring the repre-  
2280 sentation of words, one of the smallest semantic units, then moved on  
2281 to sentences, then to knowledge bases, which purpose to represent whole  
2282 pans of human knowledge. Another underlying thread to this chapter is  
2283 the notion of relationship. While the idea is quite pervasive in Section 1.4,  
2284 it is also present in Section 1.2 through the not-so-randomly chosen ex-  
2285 ample of Figure 1.1.<sup>28</sup> Even in Section 1.3, representations of sentences  
2286 are obtained by modeling the relationship of words with each other. For  
2287 example, in a transformer, the attention weights capture the relationship  
2288 between two words: the query and one element of the memory.

2289 In the next chapter, we make the link between the two symbolic rep-  
2290 resentations of meaning we studied: language and knowledge bases. More  
2291 specifically, we present relation extraction models. State-of-the-art mod-  
2292 els build heavily on the distributed representations methods introduced in  
2293 this chapter and are the main focus of this thesis.

2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322

<sup>28</sup> This figure presented the word embeddings of some countries and their capitals. The relationship between the words seems to bear the same regularity as the relationship between the underlying entities. This regularity being representative of the *capital of* relationship.

2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376

## Chapter 2

# Relation Extraction

The rapid increase in the amount of published information brings forward the problem of how to handle large amounts of data. To this goal, *information extraction* aims at discovering the underlying semantic structure of texts. As such, it is considered to be a part of natural language understanding. It is the link from unstructured text to structured data. Following Section 1.4, we will use knowledge bases as a formalization of structured data. However, to encompass the notion of information more appropriately, the concept of knowledge base needs to be taken in a broad sense. The strict definition of knowledge underlying most knowledge bases only includes general facts and does not encompass things such as “Seneca is contemptuous even of the best garum.” However, this sentence conveys a piece of information that needs to be considered by information extraction systems. As such, we will consider text-specific facts such as “Seneca dislikes garum” to be facts belonging in a knowledge base.

In this thesis, we focus on relation extraction, a subtask of information extraction. Precursors of relation extraction were the template filling tasks. In these tasks, objects corresponding to a given class—usually a specific kind of event—must be extracted from a text, and a template must be filled with information about this object. This was pioneered by Sager (1972) but started gathering interest with the message understanding conferences (MUC) supported by DARPA.<sup>29</sup> The template filling task was formalized and evaluated in a systematic way starting with MUC-2<sup>30</sup> in 1989. But it was not until 1997 that MUC-7 formalized the modern relation extraction task. The MUCs were succeeded by the automatic content extraction (ACE) program convened by the NIST<sup>31</sup> starting in 1999.

The main information extraction task is known as *knowledge base population* and consists in generating knowledge base facts from a set of documents. This task can be broken down into several steps, as illustrated by Figure 2.1:

**Entity chunking** seeks to locate entities in text. A similar task is named entity recognition (NER) which not only locates the entities but also assigns them with a type such as “organization,” “person,” “location,” etc. The relation extraction datasets we consider in subsequent chapters do not include this entity-type information. However, NER was more prevalent in relation extraction works during the 2000s decade.

**Entity linking** assigns a knowledge base entity identifier to a tagged

“When two objects, qualities, classes, or attributes, viewed together by the mind, are seen under some connexion, that connexion is called a relation.

— Augustus De Morgan, “On the Syllogism, No. III, and on Logic in general” (1864, p. 203)

“Hard constraints are the midwife to good design.

— Maciej Ceglowski, *Web Design: The First 100 Years* (2014)

In contrast to relation extraction, when filling a template about an entity, the template has a fixed number of fields to be filled, in the language of Section 1.4.1, this means that all relations are left-total:  $r \bullet \tilde{r} = r \bullet \tilde{r} \cup \mathbf{I}$ .

Sager, “Syntactic Formatting of Science Information” AFIPS 1972

<sup>29</sup> The Defense Advanced Research Projects Agency, a research agency of the USA Department of Defense.

<sup>30</sup> At the time, the conference was known as MUCK-II.

<sup>31</sup> The National Institute of Standards and Technology, an agency of the USA Department of Commerce.

2431 entity in a sentence. This disambiguates “Paris, France” Q90, from  
 2432 “Paris, son of Priam, king of Troy” Q167646 and “Paris, genus of the  
 2433 true lover’s knot plant” Q162121. Following the above discussion on  
 2434 our broad sense of knowledge, an entity may not necessarily appear  
 2435 in an existing knowledge base, in which case the entity identifier can  
 2436 be taken to be the entity’s surface form.

2437  
 2438 **Relation extraction** assigns a knowledge base relation identifier to an  
 2439 ordered pair of tagged entities in a sentence. Paris is not only the  
 2440 capital of France, it is also located in France. However, the sentence  
 2441 of Figure 2.1 does not convey the idea of location but the one of  
 2442 capital, thus predicting “*located in country*” P17 would be incorrect  
 2443 there.

2444  
 2445 Whereas Chapter 1 introduces the main tools used in relation extrac-  
 2446 tion systems, the present chapter focuses on the relation extraction task  
 2447 itself. We formally define relation extraction in Section 2.1 and introduce  
 2448 its main variants encountered in the literature. A fundamental problem  
 2449 of relation extraction models is how to obtain supervision. Hand label-  
 2450 ing a dataset is tedious and error-prone, so several alternative supervision  
 2451 techniques have been considered over the years; this is the focus of Sec-  
 2452 tion 2.2. We then introduce noteworthy supervised approaches—including  
 2453 weakly and semi-supervised ones—in Sections 2.3 and 2.4. As we will see  
 2454 in Section 2.1, the task can be tackled at the sentence level or at a higher  
 2455 level. Section 2.3 introduces sentence-level models, while Section 2.4 in-  
 2456 troduces higher-level models. Lastly, we delve into the main subject of  
 2457 this thesis, unsupervised relation extraction, in Section 2.5. Each of these  
 2458 sections is generally ordered following historical development, with older  
 2459 methods appearing first and current state-of-the-art appearing last.

2460  
 2461

## 2462 2.1 Task Definitions

2463  
 2464

2465 The relation extraction task was shaped by several datasets with different  
 2466 goals. The first MUCs focused on detecting naval sightings and engage-  
 2467 ment in military messages. Subsequent conferences moved towards the  
 2468 extraction of business-related relations in news reports. Nowadays, gen-  
 2469 eral encyclopedic knowledge is usually extracted from either news reports  
 2470 or encyclopedia pages. Another common goal is to extract drugs, chemi-  
 2471 cal and symptoms interactions in biomedical texts (Lee et al. 2019). For  
 2472 further details, Appendix C contains a list of datasets with information  
 2473 about the source of the text and the nature of the relations to be extracted.  
 2474 Depending on the end-goal for which relation extraction is used, different  
 2475 definitions of the task might be more fitting. We now formally define the  
 2476 relation extraction task and explore its popular variants.

2477 In relation extraction, we assume that information can be represented  
 2478 as a knowledge base  $\mathcal{D}_{\text{KB}} \subseteq \mathcal{E}^2 \times \mathcal{R}$  as defined in Section 1.4. In addition to  
 2479 the set of entities  $\mathcal{E}$  and the set of relations  $\mathcal{R}$ , we need to define the source  
 2480 of information from which to extract relations. The information source  
 2481 can come in several different forms, but we use a single basic definition on  
 2482 sentences which we can refine later on. We assume entity chunking was  
 2483 performed on our input data. We only deal with binary relations<sup>32</sup> since  
 2484 they are the ones commonly encoded in knowledge bases. We can therefore

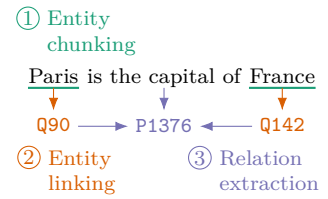


Figure 2.1: The three standard tasks for knowledge base population. First, entity chunking locates the entities in the sentence, here “Paris” and “France.” Second, entity linking map each entity to a knowledge base identifier, here Q90 and Q142. Third, relation extraction find the relation linking the two entities, here P1376 (*capital of*).

For ease of notation, we changed the placement of entities in the tuple corresponding to a fact from the one used in Section 1.4. This will allow us to refer to the entity pair as  $e \in \mathcal{E}^2$ .

<sup>32</sup> As described in Section 1.4.1, this means that only relations between two entities are considered. Moreover, higher-arity relations can be decomposed into sets of binary ones.



2485 define  $\mathcal{S}$  as a set of sentences with two tagged and ordered entities:

2486  $\mathcal{S} = \{$ “Jan Kasl <sub>$e_1$</sub>  became mayor of Prague <sub>$e_2$</sub> .”,  
 2487 “Vincent Callebaut <sub>$e_2$</sub>  was born in 1977 in Belgium <sub>$e_1$</sub> .”,  
 2488 ... }.

2491 In this example, two sentences are given; in each sentence, the relation  
 2492 we seek is the one between the two entities marked by underlines. The  
 2493 entities need to be ordered since most relations are asymmetric ( $r \neq \check{r}$ ).  
 2494 In practice, this means that one entity is tagged as  $e_1$  and the other as  
 2495  $e_2$ . The standard setting is to work on sentences; this can of course be  
 2496 generalized to larger chunks of text if needed.

2497 The tagged entities inside the sentences of  $\mathcal{S}$  are not the same as entities  
 2498 in knowledge bases. They are merely surface forms. These surface forms  
 2499 are not sensu stricto elements of  $\mathcal{E}$ . Indeed, the same entity can have  
 2500 several different surface forms, and the same surface form can be linked  
 2501 to several different entities depending on context. To map these tagged  
 2502 surface forms to  $\mathcal{E}$ , entity linking is usually performed on the corpus. In  
 2503 practice, this means that we consider samples from  $\mathcal{S} \times \mathcal{E} \times \mathcal{E}$ . Finally,  
 2504 since the two tagged entities are ordered, we simply assume that the first  
 2505 entity in the tuple corresponds to the entity tagged  $e_1$  in the sentence,  
 2506 while the second entity refers to  $e_2$ .<sup>33</sup> If entity linking is not performed  
 2507 on the dataset, we can simply assume that the surface forms are actually  
 2508 entities, in this case, and in this case alone,  $\mathcal{E}$  is a set of surface forms.  
 2509 This is somewhat uncommon, the standard practice being to have linked  
 2510 entities.

2511 Also, note that this setup is still valid for sentences with three or more  
 2512 entities, as we can consider all possible entity pairs:

2513  $\mathcal{S} = \{$ “Alonzo Church <sub>$e_1$</sub>  was born on June 14, 1903, in Washington,  
 2514 D.C. <sub>$e_2$</sub> , where his father, Samuel Robbins Church, was the judge  
 2515 of the Municipal Court for the District of Columbia.”,  
 2516 “Alonzo Church <sub>$e_2$</sub>  was born on June 14, 1903, in Washington,  
 2517 D.C., where his father, Samuel Robbins Church <sub>$e_1$</sub> , was the judge  
 2518 of the Municipal Court for the District of Columbia.”,  
 2519 ... }.

2522 In this example, we give two elements from  $\mathcal{S}$ , these elements are different  
 2523 since their markings  $\_\_e$  differ. We often use the word sentence without  
 2524 qualifications to refer to elements from  $\mathcal{S}$ . Still, even though the two sen-  
 2525 tences above are the same in the familiar sense of the term, they are  
 2526 different in our definition.

2527 Now, given a sentence with two tagged, ordered, and linked entities, we  
 2528 can state the goal of relation extraction as finding the semantic relation  
 2529 linking the two entities as conveyed by the sentence. Since the set of pos-  
 2530 sible relations is designated by  $\mathcal{R}$ , we can sum up the relation extraction  
 2531 task as finding a mapping taking the form:

$$2532 \quad \boxed{f_{\text{sentential}}: \mathcal{S} \times \mathcal{E}^2 \rightarrow \mathcal{R}} \quad (2.1)$$

2534 When we have access to a supervised dataset, all the information (head  
 2535 entity, relation, tail entity, conveying sentence) is provided. Table 2.1 gives  
 2536 some supervised samples examples. We denote a dataset of sentences with  
 2537 tagged, ordered, and linked entities as  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$  and a supervised dataset

Relation extraction can also be per-  
 formed on semi-structured documents,  
 such as a Wikipedia page with its in-  
 fobox or an HTML page that might con-  
 tain lists and tables. This is the case  
 of DIPRE presented in Section 2.3.2. As  
 long as the semi-structured data can be  
 represented as a token list, and stan-  
 dard text models can still be applied.

<sup>33</sup> Note that  $e_2$  can appears before  $e_1$   
 in the sentence.

	Head	Relation	Tail	Sentence
2539				
2540				
2541	Q210175	P159	Q198519	The exterior and interior of Freemasons’
2542	MI5	headquarters	Thames	Hall continued to be a stand-in for
2543		location	House	Thames House <sub>e<sub>2</sub></sub> , the headquarters of
2544	Q210175	P101	Q501700	MI5 <sub>e<sub>1</sub></sub> . Golitsyn’s claims about Wilson were
2545	MI5	field of work	counter-	believed in particular by the senior
2546			intelligence	MI5 <sub>e<sub>1</sub></sub> counterintelligence <sub>e<sub>2</sub></sub> officer Peter
2547				Wright. Wright, Peter (1987)
2548	Q158363	P101	Q501700	In its counter-espionage <sub>e<sub>2</sub></sub> and
2549	SMERSH	field of work	counter-	counter-intelligence roles, SMERSH <sub>e<sub>1</sub></sub>
2550			intelligence	appears to have been extremely
2551				successful throughout World War II.
2552	Q198519	P466	Q210175	The Freemasons’ Hall in London served
2553	Thames	occupant	MI5	as the filming location for Thames
2554	House			House <sub>e<sub>1</sub></sub> , the headquarters for MI5 <sub>e<sub>2</sub></sub> .

Table 2.1: Samples from the FewRel dataset. The surface forms in the head, relation and tail columns are only given for ease of reading and are usually not provided.

2557 as  $\mathcal{D}_{\mathcal{R}} \subseteq \mathcal{D} \times \mathcal{R}$ . Given an entity pair  $e = (e_1, e_2)$ , a sample in which these  
 2558 entities appear  $(s, e_1, e_2)$  is called a *mention*. A sample which convey a  
 2559 fact  $e_1 r e_2$  is called an *instance* of  $r$ .

2560 The relation extraction task as stated by Equation 2.1 is called *senten-*  
 2561 *tial extraction*. It is the traditional relation extraction setup, the sentences  
 2562 are considered one by one, and a relation is predicted for each sentence  
 2563 separately. However, information can be leveraged from the regularities of  
 2564 the dataset itself. Indeed, some facts can be repeated in multiple sentences,  
 2565 in which case a model could enforce some kind of consistency on its pre-  
 2566 dictions. Even beyond a simple consistency of the relations predicted, in  
 2567 the same fashion that a word can be defined by its context, so can an en-  
 2568 tity. This kind of regularities can be exploited by modeling a dependency  
 2569 between samples even when conditioned on the model parameters. While  
 2570 tackling relation extraction at the sentence level might be sufficient for  
 2571 some datasets, others might benefit from larger context, especially when  
 2572 the end goal is to build a knowledge base containing general facts. This  
 2573 gives rise to the *aggregate extraction* setting, in which a set of tagged sen-  
 2574 tences is directly mapped to a set of facts without a direct correspondence  
 2575 between individual sentences and individual facts.

$$2576 \quad f_{\text{aggregate}} : 2^{\mathcal{S} \times \mathcal{E}^2} \rightarrow 2^{\mathcal{E}^2 \times \mathcal{R}} \quad (2.2)$$

2579 Quite often in this case, the problem is tackled at the level of entity pairs,  
 2580 meaning that instead of making a prediction from a sample in  $\mathcal{S} \times \mathcal{E}^2$ , the  
 2581 prediction is made from  $2^{\mathcal{S} \times \mathcal{E}^2}$ . This setup is required for multi-instance  
 2582 approaches presented in Section 2.4.2. Aggregate extraction may impose  
 2583 a relatively more transductive approach<sup>34</sup> since predictions rely directly  
 2584 on previously observed samples. Usually, aggregate models still extract  
 2585 some form of prediction at the sentence level, even if they do not need to.  
 2586 Therefore, the key point of aggregate approaches is the explicit handling of  
 2587 dataset-level information. Some models may heavily depend on this global  
 2588 information, to the point that they cannot be trained without some form of  
 2589 repetition in the dataset. The sentential–aggregate distinction constitutes  
 2590 a spectrum. While all unsupervised methods exhibit some aggregate traits,  
 2591 they do not necessarily exploit as much structural information as they  
 2592 could; this is the key point of Chapter 4.

Mentions as defined here can be called “entity mentions,” while instances may be referred to as “relation mentions.”

The left-hand side of Equation 2.2 is a subset of  $\mathcal{S} \times \mathcal{E}^2$ , that is  $\mathcal{D}$  or a subset thereof. On the right-hand side, we have a subset of  $\mathcal{E}^2 \times \mathcal{R}$ ; we intend to find  $\mathcal{D}_{\text{KB}}$  or a subset thereof. However, each individual sample  $(s, e) \in \mathcal{D}$  does not need to be mapped to an individual fact  $(e, r) \in \mathcal{D}_{\text{KB}}$ .

<sup>34</sup> Transductive approaches are contrasted to inductive approaches. In the inductive approach—such as neural networks—parameters  $\theta$  are estimated from the training set. When labeling on an unknown sample, the model makes its prediction only from parameters  $\theta$  and the unlabeled sample, access to the training set is no longer necessary. This is called induction since “rules” ( $\theta$ ) are obtained from examples. On the other hand,

### 2.1.1 Nature of Relations

The supervised relation extraction task described above is quite generic. The approaches to tackle it in practice vary quite a lot depending on the specific nature of the facts we seek to extract and the corpus structure. In this subsection, we present some variations on the nature of  $\mathcal{R}$  commonly encountered in the literature.

#### 2.1.1.1 Unspecified Relation: *Other*

The set  $\mathcal{R}$  is built using a finite set of labels. These labels do not describe the relationship between all entities in all possible sentences. Indeed some entities are deemed unrelated in some sentences. A distinction is sometimes made between relation extraction and relation detection, depending on whether a relation is assumed to exist between the two entities in a sentence or not. This apparent absence of relation is often called “*other*,” since a relation between the two entities might exist but is simply not present in the relation schema considered (Hendrickx et al. 2010). In this case, we can still use the usual relation extraction setup by augmenting  $\mathcal{R}$  with the following relation:

$$other = \bigcap_{r \in \mathcal{R}} \bar{r}. \quad (2.3)$$

However note that “*other*” is not a relation like the others, it is defined by what it is not instead of being defined by what it is. This peculiarity calls for special care on how it is handled, especially during evaluation.

#### 2.1.1.2 Closed-domain Assumption

As stated above, the set  $\mathcal{R}$  is usually built from a finite set of labels such as *parent of* and *part of*. This is referred to as the *closed-domain assumption*. Another approach is to consider  $\mathcal{R}$  is not known beforehand (Banko et al. 2007). In particular open information extraction (OIE, Section 2.5.2) directly uses surface forms as relation labels. In this case, the elements of  $\mathcal{R}$  are strings of words, not defined in advance, and even potentially not-finite. We can see OIE as a preliminary task to relation extraction: the set of surface forms can be mapped to a traditional closed-set of labels. When  $\mathcal{R}$  is not known beforehand, the relation extraction problem can be called *open-domain relation discovery*. This is the usual setup for unsupervised relation extraction described in Section 2.5.

#### 2.1.1.3 Directionality and Ontology

Most relations  $r$  are not symmetric ( $r \neq \bar{r}$ ). There are several different approaches to handle this asymmetry. In the SemEval 2010 Task 8 dataset (Section C.6), the first entity in the sentence is always tagged  $e_1$ , and the second is always tagged  $e_2$ . The relation set  $\mathcal{R}$  is closed under the converse operation (Hendrickx et al. 2010):

$$\forall r \in \mathcal{R} : \bar{r} \in \mathcal{R}.$$

This is the most common setup. In this case, the relation labels incorporate the directionality; for example, the SemEval dataset contains both *cause-effect*( $e_1, e_2$ ) and *cause-effect*( $e_2, e_1$ ) depending on whether the first

in the transductive approach—such as K-NN—observations on the train set are directly transferred to test samples without first generalizing to a set of rules.

Hendrickx et al., “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals” SemEval 2010

We use the notation of Section 1.4.1 where  $\bar{r}$  refers to the complementary relation of the named relations  $r$  in the schema  $\mathcal{R}$ . Note that using the definition of relations as a set of entity pairs is not strictly correct here since two entities may be linked by a relation that is simply not conveyed by a specific sentence containing them. The underlying problem to this notational conundrum is the fact that *other* is only needed for mono-relation extraction when one and exactly one relation must be predicted for a sample; see Section 2.4.2 for an alternative. The definition given in Equation 2.3 is nonetheless fitting to the widespread distant supervision setting which we describe Section 2.2.2.

Banko et al., “Open Information Extraction from the Web” IJCAI 2007

entity appearing in the sentence is the cause or the effect. This means that given a  $r \in \mathcal{R}$  in the SemEval dataset, we can easily query the corresponding  $\check{r}$ . On the other hand, the relation set of the FewRel dataset (Section C.2) is not closed under the converse operation (Han et al. 2018). Furthermore, it is a mono-relation dataset without *other*. This means that all samples  $(s, e_1, e_2) \in \mathcal{D}$  convey a relation between  $e_1$  and  $e_2$ . Naturally, in this case, the entity tagged  $e_2$  may appear before the one tagged  $e_1$ . And indeed, for relations that do not have their converse in  $\mathcal{R}$ , the same sentence  $s$  with the tags reversed may not appear in the FewRel dataset since this would need to be categorized as  $\check{r} \notin \mathcal{R}$ .

In general, the order of  $e_1$  and  $e_2$  is not fixed. This is particularly true in the open-domain relation setup, when  $\mathcal{R}$  being unknown, can not be equipped with the converse operation. In this case, it is common to feed the samples in both arrangements: with the first entity tagged  $e_1$  and the second  $e_2$ , and the reverse: with the first entity tagged  $e_2$  and the second  $e_1$ . This can be seen as a basic data augmentation technique.

More generally, the relation set  $\mathcal{R}$  might possess a structure called a *relation ontology*. This is especially true when  $\mathcal{R}$  comes from a knowledge base such as Wikidata (Vrandečić and Krötzsch 2014). In this case,  $\mathcal{R}$  can be equipped with several operations other than the converse one. For example, Wikidata endows  $\mathcal{R}$  with a subset operation, the relation *parent organization* P749 is recorded as a subset of *part of* P361, such that  $e_1$  *parent organization*  $e_2 \implies e_1$  *part of*  $e_2$ , or using the notation of Section 1.4.1: *parent organization*  $\cup$  *part of* = *part of*.

2671

2672

### 2.1.2 Nature of Entities

The approach to tackle the relation extraction task also quite heavily depends on the nature of entities. In particular, an important distinction must be made on whether the *unique referent assumption* is postulated. This has been the case in most examples given thus far. For instance, “Alan Turing” designates a single human being, even if several people share this name; we only designate one of them with the entity Q7251 “Alan Turing.” However, this is not always the case, for example, in the following sample from the SemEval 2010 Task 8 dataset:

2683       The key <sub>$e_1$</sub>  was in a chest <sub>$e_2$</sub> .  
2684       Relation: *content-container*( $e_1, e_2$ )

2685

In this case, the entities “key” and “chest” do not always refer to the same object. The relation holds in the small world described by this sentence, but it does not always hold for every object designated by “key”. This is closely related to the fineness of entity linking. Indeed, one could link the surface form “key” above with an entity designating this specific key, but this is not always the case, as exemplified by the SemEval 2010 Task 8 dataset. This distinction is pertinent to the relation extraction task, especially in the aggregate setting. When applied to entities with a unique referent, the *content-container*( $e_1, e_2$ ) relation is  $N \rightarrow 1$  or at least transitive. However, when the unique referent assumption is false, this relation is not  $N \rightarrow 1$  anymore since several “key” entities can refer to different objects located in different containers.

The unique referent assumption is not binary; the distinction is quite fuzzy in most cases. Should the entity Q142 “France” refers both to the modern country and to the twelfth-century kingdom? What about the

Han et al., “FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation” EMNLP 2018

SemEval 2010 Task 8 is one of those datasets without entity linking, which is rather common when dealing with non-unique referents.

The aggregate setup is not necessarily contradictory with the unique referent assumption. Even though not all “keys” are in a “chest,” this fact still gives us some information about “keys,” in particular they can be in a “chest,” which is not the case of all entities.

2701 West Frankish Kingdom? How should we draw the distinction? Instead of  
 2702 categorizing the model on whether they take the unique referent assump-  
 2703 tion for granted, we should instead look at their capacity to capture the  
 2704 kind of relationship between a key and a chest as conveyed by the above  
 2705 sample.

2706 Finally, another variation of the definition of entities commonly en-  
 2707 countered in relation extraction comes from coreference resolution. Some  
 2708 datasets resolve pronouns such that in the sentence “She<sub>e</sub> died in Maryle-  
 2709 bone,” the word “she” can be considered an entity linked to Q7259 “Ada  
 2710 Lovelace” if the context in which the sentence appears supports this. In  
 2711 this case, the surface form of the entity gives little information about the  
 2712 nature of the entity. This can be problematic for models relying too heavily  
 2713 on entities’ surface forms. In particular, early relation extraction models  
 2714 did not have access to entity identifiers; at the time, pronoun entities were  
 2715 avoided altogether.

2716

2717

## 2718 2.2 The Problem of Data Scarcity

2719

2720 Ideally, a labeled dataset should be available for the source language and  
 2721 target relation domain  $\mathcal{R}$ , but alas, this is rarely the case. In particular,  
 2722 the order of  $\mathcal{R}$  can range in the thousands, in which case, accurate labeling  
 2723 is tedious for human operators. To circumvent this problem, alternative  
 2724 supervision strategies have been used.

2725 Despite the ubiquity of the terms, it is not easy to define the dif-  
 2726 ferent forms of supervision clearly. We use the following practical defini-  
 2727 tion: a dataset is supervised if among its features, one—the labels—must  
 2728 be predicted from the others. Furthermore, to distinguish with the self-  
 2729 supervised setup, we need to impose that the labels must be somewhat  
 2730 hard to obtain, typically through manual annotation.<sup>35</sup> For our task at  
 2731 hand, a supervised dataset takes the form  $\mathcal{D}_{\mathcal{R}} \subseteq \mathcal{S} \times \mathcal{E}^2 \times \mathcal{R}$ , indeed we seek  
 2732 to predict relation labels and obtaining those is tedious and error-prone.  
 2733 On the other hand, an unsupervised dataset takes the form  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$ ,  
 2734 which is much easier to obtain: vast amounts of text are now digitized and  
 2735 can be processed by an entity chunker and an entity linker. An intermedi-  
 2736 ate supervision setting is semi-supervision when a small subset of samples  
 2737 are supervised while other are left unsupervised, which can be stated as  
 2738  $\mathcal{D}_{\text{semi}} \subseteq \mathcal{S} \times \mathcal{E}^2 \times (\mathcal{R} \cup \{\varepsilon\})$ .<sup>36</sup>

2739 Despite these different kinds of datasets on which a relation extrac-  
 2740 tion model can be trained, evaluating such a model is nearly always done  
 2741 using a supervised dataset  $\mathcal{D}_{\mathcal{R}}$ . In this section, we present two other ap-  
 2742 proaches to train a model without manual labeling: bootstrap and distant  
 2743 supervision.

2744

2745

### 2746 2.2.1 Bootstrap

2747

2748 Another method to deal with the scarcity of data is to use bootstrap. Early  
 2749 approaches to relation extraction often focused on a single relation and  
 2750 fell into this category of bootstrapped methods. The bootstrap process  
 2751 (Algorithm 2.1) starts with a small amount of labeled data and finds  
 2752 extraction rules by generalizing to a large amount of unlabeled data. As  
 2753 such, it is a semi-supervised approach. We now describe this algorithm by  
 2754 following the work that pioneered this approach.

More generally, all the usual properties of grammatical nouns can lead to variations of the relation extraction task. For example, many models focus on rigid designators such as “Lucius Junius Brutus” which are opposed to flaccid designators such as “founder of the Roman Republic.” Both refer to the same person Q223440. However, it is possible to imagine a world where the “founder of the Roman Republic” does not refer to Q223440. On the contrary, if Q223440 exists, “Lucius Junius Brutus” ought to refer to him.

<sup>35</sup> To add to the confusion, the distinction between self-supervised and unsupervised is not necessarily pertinent, e.g. Yann LeCun retired “unsupervised” from his vocabulary, replacing it with “self-supervised” (LeCun and Misra 2021). In this case, the difficulty of obtaining the labels might be the sole difference between the “unsupervised/self-supervised” and “supervised” setups.

<sup>36</sup> Here, we denote by  $\varepsilon$  the absence of labels for a sample since this is often reflected by an empty field.

**algorithm** BOOTSTRAP  
*Inputs:*  $\mathcal{D}$  unlabeled dataset  
 $O$  or  $R$  seed  
*Outputs:*  $O$  occurrences  
 $R$  rules  
 Start with either  $O$  or  $R$   
**loop**  
 $O \leftarrow \{x \in \mathcal{D} \mid R \text{ matches on } x\}$   
 $R \leftarrow$  induce rules from occurrences  $O$   
**output**  $O, R$

Algorithm 2.1: The bootstrap algorithm. Occurrences are simply a set of samples  $O \subseteq \mathcal{D}$  conveying the target relation. The algorithm can be either seeded with a set of occurrences  $O$  (Brin 1999) or a set of rules  $R$  (Hearst 1992). When starting with a set of occurrences, the algorithm must first start by extracting a set of rules, then alternate between finding occurrences and rules as listed.

2755 Hearst (1992) propose a method to detect a single relation between  
 2756 noun phrases: hyponymy. They define  $e_1$  to be an hyponym of  $e_2$  when the  
 2757 sentence “An  $e_1$  is a (kind of)  $e_2$ .” is acceptable to an English speaker. This  
 2758 relation is then detected inside a corpora using lexico-syntactic patterns  
 2759 such as:<sup>37</sup>

2760  $e_1$  ,? including ( $e_2$ ,)\* (or|and)?  $e_3$   
 2761  $\Rightarrow e_2$  hyponym of  $e_1$   
 2762  $\Rightarrow e_3$  hyponym of  $e_1$

2763 where the entities  $e_i$  are constrained to be noun phrases. This rule matches  
 2764 on the following sentence:

2766 All common-law countries, including Canada and England...  
 2767  $\Rightarrow$  Canada hyponym of Common-law country  
 2768  $\Rightarrow$  England hyponym of Common-law country

2770 Hearst (1992) proposes the following process: start with known facts  
 2771 such as hyponym(England, Country), find all places where the two enti-  
 2772 ties co-occur in the corpus and write new rules from the patterns observed,  
 2773 which allows them to discover new facts to repeat the process with. Be-  
 2774 side some basic lemmatization—which explains why “countries” became  
 2775 “country” in the example above—all noun phrases are treated as possible  
 2776 entities. This is sensible since the end goal of the approach is to generate  
 2777 new facts for the WordNet knowledge base. In Hearst (1992), writing new  
 2778 rules was not done automatically but performed manually.

2779 Following equation 2.1, a sentential relation extraction system usually  
 2780 defines a relation  $r$  as a subset of  $\mathcal{S} \times \mathcal{E} \times \mathcal{E}$ , i.e. relations are conveyed  
 2781 jointly by sentences and entity pairs. In contrast, Hearst (1992) makes the  
 2782 following assumption:

2783 **Assumption  $\mathcal{H}_{\text{PULLBACK}}$ :** *It is possible to find the relation conveyed by a*  
 2784 *sample by looking at the entities alone and ignoring the sentence; and*  
 2785 *conversely by looking at the sentence alone and ignoring the entities.*

2786  $\mathcal{D} = \mathcal{S} \times_{\mathcal{R}} \mathcal{E}^2$ .

2788 This implies that given a pair of entities, whatever is the sentence in  
 2789 which they appear, the conveyed relation is the same. On the contrary,  
 2790 given a sentence, the conveyed relation is always the same, whatever the  
 2791 entities. As such the representation of a relation is split into two parts:

2793 **a set of entity pairs**  $r_{\mathcal{E}} \subseteq \mathcal{E}^2$ , which can be represented exactly;

2794 **a set of sentences**  $r_{\mathcal{S}} \subseteq \mathcal{S}$ , which in Hearst (1992) was represented by a  
 2795 set of patterns matching only sentences in  $r_{\mathcal{S}}$ , such as “ $e_1$  ,? including  
 2796 ( $e_2$ ,)\* (or|and)?  $e_3$ .”

2798 Given a dataset  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$ , it is possible to map from  $r_{\mathcal{E}}$  to  $r_{\mathcal{S}}$  by  
 2799 taking all sentences where the two entities appear and vice-versa by taking  
 2800 all pairs of entities appearing in the given sentences. The second process  
 2801  $\mathcal{R}_{\mathcal{S}} \times \mathcal{D} \rightarrow \mathcal{R}_{\mathcal{E}}$  is straightforward to implement exhaustively. While the  
 2802 first process  $\mathcal{R}_{\mathcal{E}} \times \mathcal{D} \rightarrow \mathcal{R}_{\mathcal{S}}$  was performed manually by Hearst (1992).

## 2804 2.2.2 Distant Supervision

2806 Craven and Kumlien (1999) introduced the idea of weak supervision to  
 2807 relation extraction as a compromise between hand labeled dataset and  
 2808

<sup>37</sup> The syntax used here is inspired by regular expression: “( )” are used for grouping, “?” indicates the previous atom is optional, “|” is used for alternatives and “\*” is the Kleene star meaning zero or more repetitions.

Hearst, “Automatic Acquisition of Hyponyms from Large Text Corpora” COLING 1992

The assumption of Hearst (1992) is that there are two morphisms  $\mathcal{S} \rightarrow \mathcal{R}$  and  $\mathcal{E}^2 \rightarrow \mathcal{R}$ , therefore  $\mathcal{D}$  must have a form which makes this decomposition possible:  $(s, e) \in \mathcal{D}$  if and only if  $s$  and  $e$  are mapped to the same relation. In other words,  $\mathcal{D}$  completes the two relation extraction morphisms to a commutative square:

$$\begin{array}{ccc} \mathcal{D} & \longrightarrow & \mathcal{S} \\ \downarrow & & \downarrow \\ \mathcal{E}^2 & \longrightarrow & \mathcal{R} \end{array}$$

In category theory, this object is called a pullback and noted  $\times_{\mathcal{R}}$ . This also means that given a sample from  $\mathcal{D}$ , it is possible to find its relation without looking at its sentence or its entities since either of them is sufficient.

Craven and Kumlien, “Constructing biological knowledge bases by extracting information from text sources” ISMB 1999

2809 unsupervised training. It was then popularized by Mintz et al. (2009)  
 2810 under the name *distant supervision*. Their idea is to use a knowledge base  
 2811  $\mathcal{D}_{\text{KB}} \subseteq \mathcal{E}^2 \times \mathcal{R}$  to supervise an unsupervised dataset  $\mathcal{D}$ . The underlying  
 2812 assumption can be stated as:

2813  
 2814 **Assumption  $\mathcal{H}_{\text{DISTANT}}$ :** *A sentence conveys all the possible relations be-*  
 2815 *tween all the entities it contains.*

2816  $\mathcal{D}_{\mathcal{R}} = \mathcal{D} \bowtie \mathcal{D}_{\text{KB}}$

2817 where  $\bowtie$  denotes the natural join operator:

$$2818 \quad \mathcal{D} \bowtie \mathcal{D}_{\text{KB}} = \{ (s, e_1, e_2, r) \mid (s, e_1, e_2) \in \mathcal{D} \wedge (e_1, e_2, r) \in \mathcal{D}_{\text{KB}} \}.$$

2821 In other words, each sentence  $(s, e_1, e_2) \in \mathcal{D}$  is labeled by all relations  
 2822  $r$  present between  $e_1$  and  $e_2$  in the knowledge base  $\mathcal{D}_{\text{KB}}$ . This is sometimes  
 2823 referred to as an unaligned dataset, since sentences are not aligned with  
 2824 their corresponding facts. The assumption  $\mathcal{H}_{\text{DISTANT}}$  is quite obviously false,  
 2825 and is only used to build a supervised dataset. A classifier is then trained  
 2826 on this dataset. In most works, including the one of Mintz et al. (2009), the  
 2827 model is designed to handle the vast amount of false positive in  $\mathcal{D} \bowtie \mathcal{D}_{\text{KB}}$ ,  
 2828 usually through the aggregate extraction setting (see Section 2.1).

2829 A caveat of distantly supervised datasets is that evaluation is often  
 2830 complex. Mintz et al. (2009) evaluate their approach on Freebase (Sec-  
 2831 tion C.3) by holding-out part of the knowledge base. However, the number  
 2832 of false negatives forces them to manually label the facts as true or false  
 2833 themselves.

2834

2835

## 2836 2.3 Supervised Sentential Extraction Models

2837

2838 In the supervised setup, all variables listed in Table 2.1 are given at train  
 2839 time. During evaluation, the relation must be predicted from the other  
 2840 three variables: sentence, head entity and tail entity. The predictions for  
 2841 each sample can then be compared to the gold standard.<sup>38</sup> We introduce  
 2842 the commonly used metric for evaluation on a supervised dataset in Sec-  
 2843 tion 2.3.1. The following sections focus on important supervised meth-  
 2844 ods, including weakly-supervised and semi-supervised methods. These sec-  
 2845 tions focus on sentential relation extraction methods, which realize Equa-  
 2846 tion 2.1. In contrast, Section 2.4 focuses on aggregate methods, which  
 2847 often build upon sentential approaches.

2848

2849

### 2850 2.3.1 Evaluation

2851

2852 Since supervised relation extraction is a standard multiclass classification  
 2853 task, it uses the usual  $F_1$  metric, with one small tweak to handle direction-  
 2854 ality. As for training, we use samples from  $\mathcal{D}_{\mathcal{R}} \subseteq \mathcal{S} \times \mathcal{E}^2 \times \mathcal{R}$  for evaluation.  
 2855 Let’s call  $x \in \mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$  an unlabeled sample, and  $g: \mathcal{D} \rightarrow \mathcal{R}$  the function  
 2856 which associates with each sample  $x$  its gold label in the dataset (as given  
 2857 by  $\mathcal{D}_{\mathcal{R}}$ ). Similarly, let’s call  $c: \mathcal{D} \rightarrow \mathcal{R}$  the function which associates with  
 2858 each sample  $x$  the relation predicted by the model we are evaluating. The  
 2859 standard  $F_1$  score for a relation  $r \in \mathcal{R}$  can be defined as:

2860

$$2861 \quad \text{precision}(g, c, r) = \frac{|\{x \in \mathcal{D} \mid c(x) = g(x) = r\}|}{|\{x \in \mathcal{D} \mid c(x) = r\}|} = \frac{\text{true positive}}{\text{predicted positive}}$$

2862

Mintz et al., “Distant supervision for relation extraction without labeled data” ACL 2009

The use of assumptions or modeling hypotheses noted  $\mathcal{H}_{\text{NAME}}$  is central to several relation extraction models, especially unsupervised ones. We strongly encourage the reader to look at the list of assumptions in Appendix B. The appendix provides counter-examples when appropriate. Furthermore, it lists the sections in which each assumption was introduced for reference.

<sup>38</sup> When a distant supervision dataset is used, “gold standard” is somewhat a misnomer. In this case, the relation labels are often referred to as a “silver standard” since they are not as good as possible.

$$\begin{aligned} \text{recall}(g, c, r) &= \frac{|\{x \in \mathcal{D} \mid c(x) = g(x) = r\}|}{|\{x \in \mathcal{D} \mid g(x) = r\}|} = \frac{\text{true positive}}{\text{labeled positive}} \\ F_1(g, c, r) &= \frac{2}{\text{precision}(g, c, r)^{-1} \times \text{recall}(g, c, r)^{-1}}. \end{aligned}$$

To aggregate these scores into a single number, multiple approaches are possible. First of all, micro-averaging: the true positives, predicted positive and labeled positive are averaged over all relations. In the case where all samples have one and only one label and prediction, micro-precision, micro-recall and micro- $F_1$  collapse into the same value, namely the accuracy. However, when computing a micro-metric on a dataset containing the *other* relation (Section 2.1.1.1), the samples labeled *other* are ignored, making the difference between micro-precision and micro-recall relevant again.

The second set of approaches uses macro-averaging, which means that the scores are averaged a first time for each relation before taking the average of these averages over the set of relations. This compensates for the class imbalance in the dataset since when taking the average of the averages, the score for a rare class is weighted the same as the score for a frequent class. The “directed” macro-scores are defined as usual:

$$\begin{aligned} \overrightarrow{\text{precision}}(g, c) &= \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \text{precision}(g, c, r) \\ \overrightarrow{\text{recall}}(g, c) &= \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \text{recall}(g, c, r) \\ \overrightarrow{F_1}(g, c) &= \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} F_1(g, c, r). \end{aligned}$$

However, two other variants exist. These variants try to discard the orientation of the relationship by packing together a relation  $r$  with its reverse  $\tilde{r}$ . This allows us to evaluate separately the ability of the model to find the correct relation and to find which entity is the subject ( $e_1$ ) and which is the object ( $e_2$ ). The simplest way to achieve this is to simply ignore the orientation:

$$\overleftrightarrow{\text{precision}}(g, c) = \frac{1}{|\mathcal{R}^\dagger|} \sum_{\{r, \tilde{r}\} \in \mathcal{R}^\dagger} \frac{|\{x \in \mathcal{D} \mid c(x), g(x) \in \{r, \tilde{r}\}\}|}{|\{x \in \mathcal{D} \mid c(x) \in \{r, \tilde{r}\}\}|},$$

where  $\mathcal{R}^\dagger$  is the set of relations paired by ignoring directionality. The set  $\mathcal{R}^\dagger$  is well defined, since for the datasets using this metric,  $\mathcal{R}$  is closed under the reverse operation  $\tilde{\cdot}$  with the notable exception of *other*. However, similarly to micro-metrics, *other* is often ignored altogether. It only influences the final metrics through the degradation of recall on samples mispredicted as *other* and of precision on samples mispredicted as not *other*. Following the definitions above, we can similarly define  $\overleftrightarrow{\text{recall}}$  and  $\overleftrightarrow{F_1}$ .

Finally, as a compromise between the directed  $\overrightarrow{F_1}$  and undirected  $\overleftrightarrow{F_1}$ , the half-directed metric was designed:

$$\overleftarrow{\text{precision}}(g, c) = \frac{1}{|\mathcal{R}^\dagger|} \sum_{\{r, \tilde{r}\} \in \mathcal{R}^\dagger} \frac{|\{x \in \mathcal{D} \mid g(x) \in \{r, \tilde{r}\} \wedge c(x) = g(x)\}|}{|\{x \in \mathcal{D} \mid c(x) \in \{r, \tilde{r}\}\}|}.$$

The key difference with the undirected metric is that while the prediction and gold must still be equal to  $r$  or  $\tilde{r}$ , they furthermore need to be equal



2917 to each other. Figure 2.2 gives a visual explanation using the confusion  
 2918 matrix. Note that the distinction between directed and undirected metrics  
 2919 can also apply to micro-metrics.

2920 In conclusion, the evaluation of supervised approaches varies along  
 2921 three axes:

- 2922 • Whether *other* is considered a normal relation or is only taken into  
 2923 account through degraded precision and recall on the other classes.
- 2924 • Whether the directionality of relations is taken into account.
- 2925 • Whether class imbalance is corrected through macro-aggregation.

2926 We now describe supervised relation extraction models, starting in this  
 2927 section with sentential approaches.

### 2932 2.3.2 Regular Expressions: DIPRE

2934 Dual Iterative Pattern Relation Expansion (DIPRE, Brin 1999) follows  
 2935 the bootstrap approaches (Section 2.2.1) and thus assumes  $\mathcal{H}_{\text{PULLBACK}}$ .  
 2936 Compared to Hearst (1992), DIPRE proposes a simple automation for the  
 2937  $\mathcal{R}_{\mathcal{E}} \times \mathcal{D} \rightarrow \mathcal{R}_{\mathcal{S}}$  step—the extraction of new patterns—and applies it to  
 2938 the extraction of the “*author of book*” relation. To facilitate this automa-  
 2939 tion and in contrast to Hearst (1992), it limits itself to two entities per  
 2940 patterns. DIPRE introduces the split-in-three-affixes technique illustrated  
 2941 by Figure 2.3. The entities split the text into three parts: prefix before  
 2942 the first entity, infix between the two entities and suffix after the second  
 2943 entity. This could be considered five parts with the two entities’ surface  
 2944 forms since they are not part of any of the three affixes. This split reap-  
 2945 peared in other works since, with the simplest methods assuming that the  
 2946 infix alone conveys the relation. Even in the case of DIPRE, all three affixes  
 2947 are considered, but the infix needed to match exactly, while the prefix and  
 2948 suffix could be shortened in order to make a pattern more general. All  
 2949 patterns are specific to an URL prefix, which made the algorithm pick up  
 2950 quickly on lists of books, with the algorithm also handling patterns where  
 2951 the author appeared before the title with a simple boolean marker.

2952 In order to generate new patterns, DIPRE takes all occurrences with  
 2953 the same infix and with the title and author in the same order. To avoid  
 2954 pattern which are too general they use the following approximation of the  
 2955 specificity of a pattern:

$$2956 \text{specificity}(\text{pattern}) = -\log(P(\text{pattern matches}))$$

$$2957 \approx \text{total length of the affixes.}$$

2960 When this specificity is lower than a given threshold divided by the number  
 2961 of known books it matched, the pattern was rejected. In the experiment,  
 2962 the algorithm was run on a starting set of five (author, title) facts which  
 2963 generated three patterns, one of which is given in Figure 2.3; these patterns  
 2964 produced in turn 4 047 facts. As per Hearst (1992), the algorithm was then  
 2965 iterated once again on these new facts. The second iteration introduced  
 2966 bogus facts, which were removed manually. Finally, the third iteration  
 2967 produced a total of 15 257 *author of book* facts. Brin (1999) manually  
 2968 analyzes twenty books out of these 15 257 and found that only one of them  
 2969 was not a book but an article, while four of them were obscure enough not  
 2970 to appear in the list of a major bookseller.

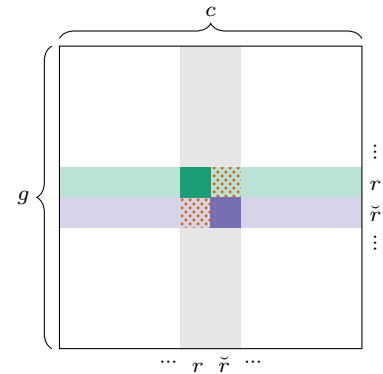


Figure 2.2: Supervised metrics defined on the confusion matrix. Directed metrics consider green and blue to be different classes, the recall for the relation  $r$  is computed by dividing the number of samples in the dark green cell by the total number of samples in the green row. Undirected metrics consider green and blue to be the same class, the recall for this class is computed by summing the four cells in the center including the two hatched ones and dividing by the sum of the two rows. Half-directed metrics also consider  $\{r, \tilde{r}\}$  to form a single class but the recall is computed by summing the two dark cells in the center—ignoring the two hatched ones—and dividing by the sum of the two rows.

Brin, “Extracting Patterns and Relations from the World Wide Web” webDB 1999

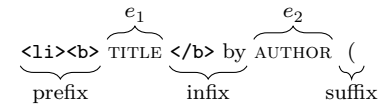


Figure 2.3: DIPRE split-in-three-affixes method. The algorithm ran on HTML code, `<li>` marks a list item, while `<b></b>` surrounds bold text.

2971 A limitation of the bootstrap approaches assuming  $\mathcal{K}_{\text{PULLBACK}}$  is that  
 2972 this assumption naively entails the following:

2973 **Assumption  $\mathcal{K}_{1\text{-ADJACENCY}}$ :** *There is no more than one relation linking any*  
 2974 *two entities.*

2975  $\forall r_1, r_2 \in \mathcal{R}: r_1 \cap r_2 = \mathbf{0}$

2976 Indeed, if a pair of entities is linked by two relations, this would implies  
 2977 a sentence containing these two entities also convey the two relations. By  
 2978 induction it follows that the two relations would actually be the same.

2979 The approach of DIPRE was subsequently used by other systems such  
 2980 as Snowball (Agichtein and Gravano 2000), which uses more complex  
 2981 matching and pattern generation algorithms and formalizes the exper-  
 2982 imental setup. We now focus on another semi-supervised approach similar  
 2983 to bootstrap, which was important to the development of relation extrac-  
 2984 tion methods.

### 2988 2.3.3 Dependency Trees: DIRT

2989 Discovery of Inference Rules from Text (DIRT, D. Lin and Pantel 2001)  
 2990 also uses the  $\mathcal{K}_{\text{PULLBACK}}$  assumption but makes a single iteration of the  
 2991 bootstrap algorithm from a single example. Furthermore, DIRT makes the  
 2992 pattern building  $\mathcal{R}_{\mathcal{E}} \times \mathcal{D} \rightarrow \mathcal{R}_{\mathcal{S}}$  more resilient to noise and applies the  
 2993 algorithm to multiple relations. Another difference is that it factorizes the  
 2994 definition of  $\mathcal{R}_{\mathcal{S}}$  using dependency paths instead of regular expressions.  
 2995 Given a sentence, a dependency parser can create a tree where nodes  
 2996 are built from words, and the arcs between the nodes correspond to the  
 2997 grammatical relationship between the words. This is called a dependency  
 2998 tree and is exemplified by Figure 2.4. After building a dependency tree,  
 2999 we can take the path between two nodes in the tree, for example the path  
 3000 between “John” and “problem” in the tree of Figure 2.4 is:

3001  $\leftarrow \text{N} : \text{subj} : \text{V} \leftarrow \text{find} \rightarrow \text{V} : \text{obj} : \text{N} \rightarrow \text{solution} \rightarrow \text{N} : \text{to} : \text{N} \rightarrow$

3002 Note that lemmatization is performed on the nodes. D. Lin and Pantel  
 3003 (2001) state their assumption as an extension of the distributional hy-  
 3004 pothesis (see section 1.1):

3005 **Distributional Hypothesis on Dependency Paths:** *If two depen-*  
 3006 *gency paths occur in similar contexts, they tend to convey similar mean-*  
 3007 *ings.*

3008 In the case of DIRT, context is defined as the two endpoints of the paths. For  
 3009 example, the context of the path given above in Figure 2.4 consists of the  
 3010 words “John” and “problem.” As such, this can be seen as a probabilistic  
 3011 version of the  $\mathcal{R}_{\mathcal{E}} \times \mathcal{D} \rightarrow \mathcal{R}_{\mathcal{S}}$  step. In order to ensure these paths correspond  
 3012 to meaningful relations, only paths between nouns are considered. For  
 3013 example, by counting all entities appearing at the endpoints of the path  
 3014 above, D. Lin and Pantel (2001) observe that the following path have  
 3015 similar endpoints:

3016  $\leftarrow \text{N} : \text{subj} : \text{V} \leftarrow \text{solve} \rightarrow \text{V} : \text{obj} : \text{N} \rightarrow$

3017 Therefore, they can conclude that these two paths correspond to the same  
 3018 relation. The orientation of a path is not essential. If the subject of “solve”  
 3019 appears after its object in a sentence, we still want this path to be counted

As a reminder from Section 1.4.1:  $\mathbf{0}$  denotes the empty relation linking no entities together. So  $r_1 \cap r_2 = \mathbf{0}$  should be understood as “if we take the relation linking together all the entity pairs connected at the same time ( $\cap$ ) by  $r_1$  and  $r_2$ , we should obtain the relation linking no entities together ( $\mathbf{0}$ ).”

D. Lin and Pantel, “DIRT – Discovery of Inference Rules from Text” KDD 2001

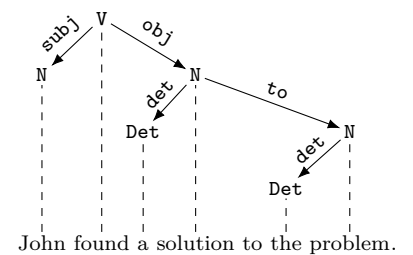


Figure 2.4: Example of dependency tree given by D. Lin and Pantel (2001) generated using the Minipar dependency parser. The nodes correspond to words in the sentence, as indicated by the dashed line. Each node is tagged by the part-of-speech (POS) of the associated word. The arrows between the nodes are labeled with the dependency between the words. The following abbreviations are used: N is noun, V is verb, Det is determiner, subj is subject, obj is object, and det is the determiner relation.

“While hunting in Africa, I shot an elephant in my pajamas. How he got into my pajamas, I don’t know.”

— Groucho Marx, Animal Crackers (1930)

The ambiguity of the prepositional phrase “in my pajamas” would be removed by a dependency tree. It can either be linked to the noun “elephant” or to the verb “shot.”

the same as the one above. As introduced in Section 2.1.1.3, this is a common problem in relation extraction. To solve this in a relatively straightforward manner, we simply assume all paths come in the two possible orientations, so for each sentence, the extracted path and its reverse are added to the dataset. We use a mutual information-based measure to evaluate how similar two set of endpoints are. Since counting all possible pairs would be too memory intensive—the squared size of the vocabulary  $|V|^2$  is usually in the order of the billion or more—we measure the similarity of the first and second endpoint separately. To measure the preference of the dependency path  $\pi$  to have the word  $w \in V$  appears at the endpoint  $\ell \in \{\leftarrow, \rightarrow\}$ , the following conditional pointwise mutual information is used:

$$\begin{aligned} \text{pmi}(\pi, w | \ell) &= \log \frac{P(\pi, w | \ell)}{P(\pi | \ell)P(w | \ell)} \\ &= \log \frac{P(\pi, \ell, w)P(\ell)}{P(\pi, \ell)P(\ell, w)}. \end{aligned}$$

This quantity can be computed empirically using a hash table counting how many time the triplet  $(\pi, \ell, w)$  appeared in the dataset. We can then compute the similarity between two paths given an endpoint  $\ell$  then take the geometric average for the two possible value of  $\ell$  to obtain an unconditioned similarity between paths:

$$\begin{aligned} \text{sim}(\pi_1, \pi_2, \ell) &= \frac{\sum_{w \in C(\pi_1, \ell) \cap C(\pi_2, \ell)} (\text{pmi}(\pi_1, w | \ell) + \text{pmi}(\pi_2, w | \ell))}{\sum_{w \in C(\pi_1, \ell)} \text{pmi}(\pi_1, w | \ell) + \sum_{w \in C(\pi_2, \ell)} \text{pmi}(\pi_2, w | \ell)} \\ \text{sim}(\pi_1, \pi_2) &= \sqrt{\text{sim}(\pi_1, \pi_2, \leftarrow) \times \text{sim}(\pi_1, \pi_2, \rightarrow)}, \end{aligned}$$

where  $C(\pi, \ell)$  designates the context, that is the set of words appearing at the endpoint  $\ell$  of the path  $\pi$ .

Using this similarity function, D. Lin and Pantel (2001) can find sets of paths corresponding to particular relations by looking at frequent paths above a fixed similarity threshold. They evaluate their method manually on a question answering dataset. For each question, they extract the corresponding path and then look at the 40 most similar paths in their dataset and manually tag whether these paths would answer the original question. The accuracy of DIRT ranges from 92.5% for the relation “*manufactures*” to 0% for the relation “*monetary value of*” for which no similar paths were found.

### 2.3.4 Hand-designed Feature Extractors

The first supervised systems for relation extraction were designed for the template relations (TR) task of the seventh message understanding conference (MUC-7). The best result was obtained by the IE<sup>2</sup> system (Aone et al. 1998), which relied on manual pattern development, with an  $F_1$  score of 76%. A close second was the 71%  $F_1$  score of the SIFT system (S. Miller et al. 1998), which was devoid of hand-written patterns. SIFT builds an augmented parse tree of the sentence, where nodes are added to encode the semantic information conveyed by each constituent. New nodes are created using an algorithm akin to a probabilistic context-free grammar using maximum likelihood. The semantic annotations are chosen following co-occurrence counts in the training set, using dynamic programming

The similarity metric equations in D. Lin and Pantel (2001) are quite informal. In particular, they do not state that  $\ell$  has a special role as a conditional variable in the pmi and erroneously designate the same value as  $\text{mi}(\pi, m, \ell)$ . The equations given here are our own.

To put these results into perspective with latter work, note that Aone et al. (1998) mention they ran their model a 167 MHz processor with 128 MB of RAM. S. Miller et al., “BBN: Description of the SIFT System as Used for MUC-7” MUC 1998

3079 to search the space of augmented parse trees efficiently. SIFT also uses  
 3080 a model to find cross-sentence relations, which represent 10–20% of the  
 3081 test set. The predictions are made from a set of elemental features, one of  
 3082 which was whether the candidate fact was seen in a previous sample; this  
 3083 gives a slight aggregate orientation to SIFT, even though it is primarily a  
 3084 sentential approach (Section 2.1). This first systematic evaluation of mod-  
 3085 els on the same dataset set the stage for the development of the relation  
 3086 extraction task.

3087 Subsequently, several methods built upon carefully designed features.  
 3088 This is for example the case of Kambhatla (2004) who use the maximum  
 3089 entropy principle on the following set of features:

- 3090 • entities and infix words with positional markers,
- 3091
- 3092 • entity types by applying NER to the corpus,
- 3093
- 3094 • entity levels, that is whether the entity is a composite noun or a pro-  
 3095 noun which was linked to an entity through coreference resolution,
- 3096
- 3097 • the number of other words and entities appearing between  $e_1$  and  
 3098  $e_2$ ,
- 3099 • whether  $e_1$  and  $e_2$  are in the same noun phrase, verb phrase or  
 3100 prepositional phrase,
- 3101
- 3102 • the dependency neighborhood, that is the neighboring nodes in the  
 3103 dependency tree (see Figure 2.4),
- 3104
- 3105 • the syntactic path, that is the path between the entities in the syn-  
 3106 tactic parse tree (see Figure 2.5).

3107 Let’s call  $(f_i(x, r))_{i \in \{1, \dots, n\}}$  the indicator functions which equal 1 iff  $x$  has  
 3108 feature  $i$  and convey  $r$ . The maximum entropy principle states that a  
 3109 classifier should match empirical data on the observed space but should  
 3110 have maximal entropy outside it. Calling  $Q^*$  the optimal probability model  
 3111 in this sense, we have:

$$\begin{aligned}
 3112 \quad Q^* &= \operatorname{argmax}_{Q \in \mathcal{Q}} H(Q) \\
 3113 &= \operatorname{argmax}_{Q \in \mathcal{Q}} \sum_{(x, r) \in \mathcal{D}} -Q(x, r) \log Q(r | x) \\
 3114 &= \operatorname{argmax}_{Q \in \mathcal{Q}} \sum_{(x, r) \in \mathcal{D}} -\hat{P}(x) Q(r | x) \log Q(r | x),
 \end{aligned}$$

3119 where  $\mathcal{Q}$  is the set of probability mass functions matching observations:

$$3120 \quad \mathcal{Q} = \left\{ \text{p.m.f. } Q \mid \mathbb{E}_{(x, r) \sim Q} [f_i(x, r)] = \mathbb{E}_{(x, r) \sim \hat{P}} [f_i(x, r)] \right\}.$$

3123 Given this setup, the solution is part of a very restricted class of functions:

$$3124 \quad Q^*(r | x; \boldsymbol{\lambda}) \propto \exp \sum_{i=1}^n \lambda_i f_i(x, r).$$

3128 The parameters  $\boldsymbol{\lambda}$  are estimated using an algorithm called generalized  
 3129 iterative scaling (GIS, Darroch and Ratcliff 1972). Using this approach,  
 3130 Kambhatla (2004) evaluate their model on a dataset succeeding MUC-7  
 3131 called ACE (to be precise, ACE 2003, see Section C.1 for details). They  
 3132 achieve an  $F_1$  of 52.8% on 24 ACE relation subtypes.

Kambhatla, “Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction” ACL 2004

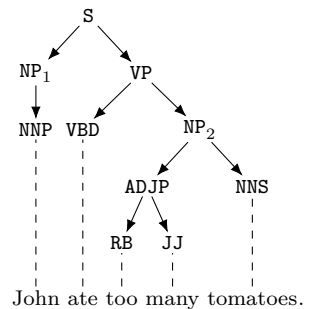


Figure 2.5: Example of syntactic parse tree generated by the PCFG parser (Klein and Manning 2003). The following abbreviations are used: S (simple declarative clause), NP (noun phrase), VP (verb phrase), ADJP (adjective phrase), NNS (plural noun), NNP (singular proper noun), RB (adverb), JJ (adjective). In contrast to a dependency tree (Figure 2.4), the words correspond to the tree’s leaves, while internal nodes correspond to constituents clauses.

As a reminder,  $\hat{P}$  denotes the empirical distribution.

### 2.3.5 Kernel Approaches

Designing a set of low-dimensional features is a tedious task: a large set of features can be computationally prohibitive, while a small set of features is necessarily limiting since they can never completely capture the essence of all samples which live in higher dimension. The kernel approaches seek to avoid this limitation by comparing samples pairwise without passing through an explicit intermediary representation. To do so, a kernel function  $k$  is defined over pair of samples:

$$k: (\mathcal{S} \times \mathcal{E}^2) \times (\mathcal{S} \times \mathcal{E}^2) \rightarrow \mathbb{R}_{\geq 0},$$

where  $k$  acts as a similarity measure and is required to be symmetric and positive-semidefinite. It can be shown that there is an equivalence between kernel functions and features space; for each kernel function  $k$  there is an implicit set of features  $\mathbf{f}$  such that  $k(x_1, x_2) = \mathbf{f}(x_1) \cdot \mathbf{f}(x_2)$ . However, some kernel function  $k$  might be computed without having to enumerate all features  $\mathbf{f}$ .

This property is used for relation extraction by Zelenko et al. (2003) who define a similarity function  $k$  between shallow parse trees.<sup>39</sup> The tree kernel is defined through a similarity on nodes with a recursive call on children nodes. The equivalent feature space would need to contain all possible sub-trees which are impractical to enumerate. Zelenko et al. (2003) train a support vector machine (SVM, Cortes and Vapnik 1995) and a voted perceptron (Freund and Schapire 1999) on a dataset they hand-labeled. Culotta and Sorensen (2004) used a similar approach with a tree kernel, except that they used dependency trees (Figure 2.4) instead of syntactic parse trees. They trained SVMs on the ACE 2004 dataset (Section C.1), with their best setup reaching an  $F_1$  of 63.2%. Finally, Zhou et al. (2005) also trained an SVM but directly used the dot product inside the feature space as a kernel.<sup>40</sup> Extracting a wide variety of features, they were able to reach an  $F_1$  score of 74.7% on the ACE 2004 dataset.

### 2.3.6 Piecewise Convolutional Neural Network

In the 2010s, machine learning models moved away from hand-designed features towards automatic feature extractors (Section 1.1). In relation extraction, this move was initiated by Socher et al. (2012) using an RNN-like model (Section 1.3.2), but it really started to gain traction with piecewise convolutional neural networks (PCNN, Zeng et al. 2015). PCNNs perform supervised relation extraction using deep learning. In contrast to previous models, they learn a CNN feature extractor (Section 1.3.1) on top of word2vec embeddings (Section 1.2.1) instead of using hand-engineered features. Furthermore, PCNN uses the split-in-three-affixes method of DIPRE (Figure 2.3). They feed each affix to a CNN followed by a max-pooling to obtain a fixed-length representation of the sentence, which depends on the position of the embeddings. This representation is then used to predict the relation using a linear and softmax layer. While the global position invariance of CNN is interesting for language modeling, phrases closer to entities might be of more importance for relation extraction, thus PCNN also uses temporal encoding (Section 1.3.3.2). Figure 2.6 showcases a PCNN model.

The setup described above can be used for sentential relation extraction. However, Zeng et al. (2015) and subsequent works place themselves

Zelenko et al., “Kernel Methods for Relation Extraction” JMLR 2003

<sup>39</sup> A shallow parse tree is similar to a syntactic parse tree (Figure 2.5) on a partition of the words of a sentence (S. P. Abney 1991).

Culotta and Sorensen, “Dependency Tree Kernels for Relation Extraction” ACL 2004

Zhou et al., “Exploring Various Knowledge in Relation Extraction” ACL 2005

<sup>40</sup> In the same way that a kernel always corresponds to the dot product in a feature space, the reverse can be shown to be true too, since a Gram matrix is always semidefinite positive.

Zeng et al., “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks” EMNLP 2015



the entity surface form (Section 2.1.2 describes why this could be desirable).

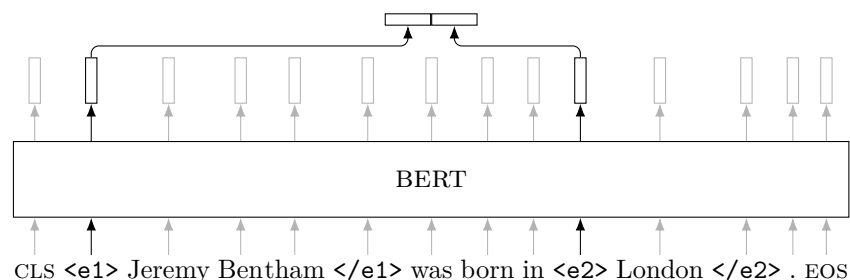


Figure 2.7: MTB entity markers–entity start sentence representation. “Bentham” was split into two subword tokens, “Ben-” and “-tham” by the BPE algorithm described in Section 1.2.3. The contextualized embeddings of most words are ignored. The final representation is only built using the representation of <e1> and <e2>. However, note that these representations are built from all the words in the sentence using an attention mechanism (Section 1.3.3). In the original work of Soares et al. (2019), the representation extracted by BERT is either fed through layer normalization (Ba et al. 2016) or to a linear layer depending on the dataset.

The best results obtained by MTB were with the entity markers–entity start method. This is the method we focus on from now on. We refer to this sentence representation model by the function  $\text{BERTcoder}: \mathcal{S} \rightarrow \mathbb{R}^d$  illustrated Figure 2.7. Training is performed using a softmax layer of size  $|\mathcal{R}|$  with a cross-entropy loss. Using a standard BERT-large pre-trained on a MLM task, MTB obtains a macro- $\overline{F}_1$  of 89.2% on the SemEval 2010 Task 8 (Section C.6).

## 2.4 Supervised Aggregate Extraction Models

All the approaches introduced thus far are sentential. They map each sample to a relation individually, without modeling the interactions between samples. In contrast, this section focuses on aggregate approaches (Equation 2.2). Aggregate approaches explicitly model the connections between samples. The most common aggregate method is to ensure the consistency of relations predicted for a given entity pair  $e \in \mathcal{E}^2$  by processing together all sentences  $s \in \mathcal{S}$  mentioning  $e$ . To this end, we define  $\mathcal{D}^e$  to be the dataset  $\mathcal{D}$  grouped by entity pairs. Thus, instead of containing a sample  $x = (s, e)$ , the dataset  $\mathcal{D}^e$  contains bag of mentions  $\mathbf{x} = \{(s, e), (s', e), \dots\}$  of the same entity pair  $e$ . Most aggregate methods are built upon sentential approaches and provide a sentential assignment. Therefore, more often than not, each sample is still mapped to a relation. Therefore, the evaluations of aggregate methods follow the evaluations of sentential approaches introduced in Section 2.3.1.

### 2.4.1 Label Propagation

To deal with the shortage of manually labeled data, one approach is to use labels weakly correlated with the samples as in distant supervision (Section 2.2.2). Another approach is to label a small subset of the dataset but leave most samples unlabeled. This is the semi-supervised approach. The bootstrapped models (Section 2.2.1) can also be seen as semi-supervised approaches: a small number of labeled samples are given to the model, which then crawls the web to obtain new unsupervised samples. The evaluation of semi-supervised models follows the one of supervised models described in Section 2.3.1. The difference between the two lies in the fact that unsupervised samples can be used to gain a better estimate of the input distribution in the semi-supervised settings, while fully-supervised models cannot make use of unsupervised samples.

3295 Apart from bootstrapped models, one of the first semi-supervised re-  
 3296 lation extraction systems was proposed by Chen et al. (2006). They build  
 3297 their model on top of hand-engineered features (Section 2.3.4) compared  
 3298 using a similarity function. This is somewhat similar to kernel approaches  
 3299 (section 2.3.5), except that this function does not need to be positive  
 3300 semidefinite. Given all samples in feature space, the labels from the super-  
 3301 vised samples are propagated to the neighboring unlabeled samples using  
 3302 the label propagation algorithm (X. Zhu and Ghahramani 2002) listed as  
 3303 Algorithm 2.2. This propagation takes the form of a convex combination  
 3304 of other samples’ labels weighted by the similarity function. Let’s call sim  
 3305 this unlabeled sample similarity function:

$$3306 \quad \text{sim}: (\mathcal{S} \times \mathcal{E}^2) \times (\mathcal{S} \times \mathcal{E}^2) \rightarrow \mathbb{R}.$$

3308 The label propagation algorithm builds a pairwise similarity matrix be-  
 3309 tween labeled and unlabeled samples which have been column normalized  
 3310 then row normalized:

$$3311 \quad t_{ij} \propto \frac{\exp(\text{sim}(x_i, x_j))}{\sum_{x_k \in \mathcal{D} \cup \mathcal{D}_{\mathcal{X}}} \exp(\text{sim}(x_k, x_j))} \quad \text{for } i, j \in \{1, \dots, |\mathcal{D}| + |\mathcal{D}_{\mathcal{X}}|\} \quad (2.4)$$

3315 The relation assigned to each unlabeled sample is then recomputed by  
 3316 aggregating the labels—whether these labels come from  $\mathcal{D}_{\mathcal{X}}$  or were com-  
 3317 puted at a previous iteration—of all other samples weighted by  $\mathbf{T}$ . Note  
 3318 that labels assigned to samples coming from  $\mathcal{D}_{\mathcal{X}}$  are not altered. This op-  
 3319 eration is repeated until the label assignment stabilizes. This label propa-  
 3320 gation algorithm has been shown to converge to a unique solution (X. Zhu  
 3321 and Ghahramani 2002).

3322 Chen et al. (2006) tried two similarity functions: the cosine and the  
 3323 Jensen–Shannon of the feature vectors. They evaluated their approach on  
 3324 the ACE 2003 dataset (Section C.1) using different fractions of the labels  
 3325 to show that while their model was roughly at the same performance level  
 3326 than others when using the whole dataset, it decisively outperformed other  
 3327 methods when using a small number of labels.

3328

## 3329 2.4.2 Multi-instance Multi-label

3331 Following the popularization of distant supervision by Mintz et al. (2009),  
 3332 training datasets gained in volume but lost in quality (see Section 2.2.2). In  
 3333 order to create models more resilient to the large number of false-positive  
 3334 in distantly-supervised datasets, multi-instance approaches (Dietterich et  
 3335 al. 1997) started to get traction.

3336 In the article of Mintz et al. (2009), all mentions of the same entity pair  
 3337 are viewed as a single sample to make a prediction. Their model is a simple  
 3338 logistic classifier on top of hand-engineered features, which could only  
 3339 predict a single relation label per entity pair. However, when aggregating  
 3340 the features of all mentions and supervising with a single relation, Mintz  
 3341 et al. (2009) backpropagate to all features, i.e. the parameters used by all  
 3342 mentions are modified. This assumes that all mentions should convey the  
 3343 relation. To avoid this assumption, the more sophisticated multi-instance  
 3344 assumption is used:

3345 **Assumption  $\mathcal{H}_{\text{MULTI-INSTANCE}}$ :** All facts  $(e, r) \in \mathcal{D}_{\text{KB}}$  are conveyed by at  
 3346 least one sentence of the unlabeled dataset  $\mathcal{D}$ .

$$3348 \quad \forall (e_1, e_2, r) \in \mathcal{D}_{\text{KB}} : \exists (s, e_1, e_2) \in \mathcal{D} : (s, e_1, e_2) \text{ conveys } e_1 r e_2$$

Chen et al., “Relation Extraction Using Label Propagation Based Semi-Supervised Learning” ACL 2006

### algorithm LABEL PROPAGATION

```

Inputs:  $\mathcal{D}_{\mathcal{X}}$  labeled dataset
         $\mathcal{D}$  unlabeled dataset
Output:  $\hat{r}$  relation predictions

▷ Initialization ◁
 $\mathbf{T} \leftarrow$  computed using Equation 2.4
    from  $\mathcal{D}_{\mathcal{X}}$  and  $\mathcal{D}$ 
 $\mathbf{Y} \leftarrow$  random stochastic matrix
for all  $(s_i, e_i, r_i) \in \mathcal{D}_{\mathcal{X}}$  do
  |  $y_{ij} \leftarrow \delta_{j,r_i}$  ◁
▷ Training ◁
loop
  |  $\mathbf{Y} \leftarrow \mathbf{T}\mathbf{Y}$ 
  | for all  $(s_i, e_i, r_i) \in \mathcal{D}_{\mathcal{X}}$  do
  | |  $y_{ij} \leftarrow \delta_{j,r_i}$ 
 $\hat{r}_i \leftarrow \text{argmax}_j y_{ij}$ 
output  $\hat{r}$ 

```

Algorithm 2.2: The label propagation algorithm. The notation  $\delta_{a,b}$  is a Kronecker delta, equals to 1 if  $a = b$  and to 0 otherwise. The two loops assigning to  $y_{ij}$  are simply enforcing that the relation assigned to the labeled samples do not deviate from their gold value.



3349 MultiR (Hoffmann et al. 2011) follows such a multi-instance setup but  
 3350 also models multiple relations and thus does not assume  $\mathcal{H}_{1\text{-ADJACENCY}}$ , un-  
 3351 like all the models introduced thus far. Figure 2.8 illustrates this setup,  
 3352 which is dubbed MIML (multi-instance multi-label) following the subse-  
 3353 quent work of Surdeanu et al. (2012).

3354 MultiR uses a latent variable  $z$  to capture the sentential extraction.  
 3355 That is, for each sentence  $x_i \in \mathcal{D}_{\mathcal{R}}$ , the latent variable  $z_i \in \mathcal{R}$  captures  
 3356 the relation conveyed by  $x_i$ . Furthermore, for a given entity pair  $e \in \mathcal{E}^2$ ,  
 3357 for all  $r \in \mathcal{R}$ , a binary classifier  $y_r$  is used to predict whether this pair  
 3358 is linked by  $r$ . In this fashion, multiple relations can be predicted for the  
 3359 same entity pair. The model can be summarized by the plate diagram of  
 3360 Figure 2.9. Let’s define  $\mathcal{D}_{\mathcal{R}}^e$  the dataset  $\mathcal{D}_{\mathcal{R}}$  where samples are grouped  
 3361 by entity pairs. Since multiple relations can link the same entity pair, we  
 3362 will use  $\mathbf{y} \in \{0, 1\}^{\mathcal{R}}$  to refer to the binary vector indexing the conveyed  
 3363 relations. Formally, MultiR defines the probability of the sentential ( $\mathbf{z}$ )  
 3364 and aggregate ( $\mathbf{y}$ ) assignments for a mention bag ( $\mathbf{x}$ ) as follow:

$$3365 \quad P(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta}) \propto \prod_{r \in \mathcal{R}} \phi^{\text{join}}(y_r, \mathbf{z}) \prod_{x_i \in \mathbf{x}} \phi^{\text{extract}}(z_i, x_i; \boldsymbol{\theta}) \quad (2.5)$$

3368 where  $\phi^{\text{join}}$  simply aggregate the predictions for all mentions:

$$3370 \quad \phi^{\text{join}}(y_r, \mathbf{z}) = \begin{cases} 1 & \text{if } y_r = 1 \wedge \exists i : z_i = r \\ 0 & \text{otherwise} \end{cases}$$

3374 and  $\phi^{\text{extract}}$  is a weighted sum of several hand-designed features:

$$3375 \quad \phi^{\text{extract}}(z_i, x_i; \boldsymbol{\theta}) = \exp \left( \sum_{\text{feature } j} \theta_j \phi_j(z_i, x_i) \right)$$

3379 We now describe the training algorithm used by MultiR, which is  
 3380 listed as Algorithm 2.3. Following the multi-instance setup, MultiR as-  
 3381 sumes that every fact  $(e_1, r, e_2) \in \mathcal{D}_{\text{KB}}$  is conveyed by at least one mention  
 3382  $(s, e_1, e_2) \in \mathcal{D}$ . This can be seen in the first product of Equation 2.5: if  
 3383 a single gold relation is not predicted for any sentence, the whole prob-  
 3384 ability mass function drops to 0. This means that during inference, each  
 3385 relation  $r$  conveyed in the knowledge base must be covered by at least one  
 3386 sentential extraction  $z$ . Given all sentences  $\mathbf{x}_i \subseteq \mathcal{D}$  containing an entity  
 3387 pair  $(e_1, e_2)$ , when the model does not predict the actual set of relations  
 3388  $\mathbf{y}_i = \{r \mid (e_1, r, e_2) \in \mathcal{D}_{\text{KB}}\}$ , the parameters  $\boldsymbol{\theta}$  must be tuned such that  
 3389 every relation  $r \in \mathbf{y}_i$  is conveyed by at least one sentence, as expressed by  
 3390 the line:

$$3391 \quad \mathbf{z}^* \leftarrow \underset{\mathbf{z}}{\text{argmax}} P(\mathbf{z} \mid \mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\theta}).$$

3393 This can be reframed as a weighted edge-cover problem, where the edge  
 3394 weights are given by  $\phi^{\text{extract}}(z_i, x_i; \boldsymbol{\theta})$ . The MultiR training algorithm can  
 3395 be seen as maximizing the likelihood  $P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$  where a Viterbi approxi-  
 3396 mation was used—the expectations being replaced with maxima.

3397 The multi-instance multi-label (MIML) phrase was introduced by Sur-  
 3398 deanu et al. (2012). Their approach is similar to that of MultiR except that  
 3399 they train a classifier for  $\phi^{\text{join}}$  instead of using a deterministic process.  
 3400 Their training procedure also differs. They train in the Bayesian frame-  
 3401 work using an expectation–maximization algorithm. In general, MIML ap-  
 3402 proaches are challenging to evaluate systematically since they suffer from

Hoffmann et al., “Knowledge-Based Weak Supervision for Information Ex-  
 traction of Overlapping Relations” ACL 2011

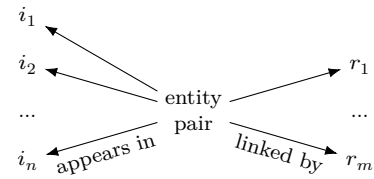


Figure 2.8: Multi-instance ( $n > 1$ ) multi-label ( $m > 1$ ) setup. Each entity pair appears in several instances and the two entities are linked by several relations.

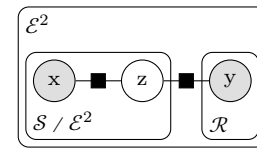


Figure 2.9: MultiR plate diagram. Where  $\blacksquare$  denotes factor nodes.

In particular, note that if an entity pair is linked by more relations than it has mentions in the text, the algorithm collapses since each mention conveys a single relation.

Surdeanu et al., “Multi-instance Multi-label Learning for Relation Extraction” EMNLP 2012

```

3403 algorithm MULTIR
3404   Input:  $\mathcal{D}_{\mathcal{R}}^e$  a supervised multi-instance dataset
3405   Output:  $\theta$  model parameters
3406    $\theta \leftarrow \mathbf{0}$ 
3407   loop
3408     for all  $(x_i, y_i) \in \mathcal{D}_{\mathcal{R}}^e$  do
3409        $(y', z') \leftarrow \underset{y, z}{\operatorname{argmax}} P(y, z \mid x_i; \theta)$ 
3410       if  $y' \neq y_i$  then
3411          $z^* \leftarrow \operatorname{argmax} P(z \mid x_i, y_i; \theta)$ 
3412          $\theta \leftarrow \theta + \underset{z}{\phi}(x_i, z^*) - \phi(x_i, z')$ 
3413       end if
3414     end for
3415   output  $\theta$ 
3416 
```

Algorithm 2.3: The MultiR training algorithm. For each bag of mentions  $x_i$ , the more likely sentential and aggregate predictions  $(y', z')$  are made. If the predicted relations are different from the true relations  $y_i$  linking the two entities, the parameters  $\theta$  are adjusted such that  $z$  cover all relations in  $y_i$ .

low precision due to incomplete knowledge bases. In particular, they were not compared with traditional supervised approaches. For reference, Surdeanu et al. (2012) compare the three methods mentioned in this section on the same datasets and observe that at the threshold at which recall goes over 30%, the precision falls under 30%.

3422

3423

### 3424 2.4.3 Universal Schemas

3425

Another important weakly-supervised model is the universal schema approach designed by Riedel et al. (2013). In their setting, existing relations and surface forms linking two entities are considered to be of the same nature. Slightly departing from their terminology, we refer to the union of relations ( $\mathcal{R}$ ) and surface forms ( $\mathcal{S}$ ) by the term “items” ( $\mathcal{J} = \mathcal{R} \cup \mathcal{S}$ ) for their similarity with the collaborative filtering concept. Riedel et al. (2013) consider that entity pairs are linked by items such that the dataset available could be referred to as  $\mathcal{D}_{\mathcal{J}} \subseteq \mathcal{E}^2 \times \mathcal{J}$ . This can be obtained by taking the union of an unlabeled dataset  $\mathcal{D}$  and a knowledge base  $\mathcal{D}_{\text{KB}}$ . This dataset  $\mathcal{D}_{\mathcal{J}}$  can be seen as a matrix with entity pairs corresponding to rows and items corresponding to columns. With this in mind, relation extraction resembles collaborative filtering. Figure 2.10 gives an example of this matrix that we will call  $M \in \mathbb{R}^{\mathcal{E}^2 \times \mathcal{J}}$ .

3439

3440

3441

3442

3443

3444

3445

3446

3447

3448

3449

3450

3451

3452

3453

3454

3455

3456

Entity pairs	“ $e_1$ professor at $e_2$ ”				“ $e_1$ historian at $e_2$ ”				$e_1$ employee of $e_2$				$e_1$ member of $e_2$			
	Surface forms				Relations											
Ferguson <sub><math>e_1</math></sub>																
Harvard <sub><math>e_2</math></sub>					1				1				1			
Oman <sub><math>e_1</math></sub>																
Oxford <sub><math>e_2</math></sub>					1				1							
Firth <sub><math>e_1</math></sub>																
Oxford <sub><math>e_2</math></sub>					0.95				1				0.97			
Gödel <sub><math>e_1</math></sub>																
Princeton <sub><math>e_2</math></sub>					1				0.05				0.93			

Riedel et al., “Relation Extraction with Matrix Factorization and Universal Schemas” NACL 2013

Figure 2.10: Universal schema matrix. Observed entity–item pairs are shown in green, blue cells are unobserved values, while orange cells are unobserved values for which a prediction was made. The observed values on the left (surface forms) come from an unsupervised dataset  $\mathcal{D}$ , while the observed values on the right (relations) come from a knowledge base  $\mathcal{D}_{\text{KB}}$ .

Riedel et al. (2013) purpose to model this matrix using a combination of three models. One of them being a low-rank matrix factorization:

$$m_{ei}^F = \sum_{j=0}^d u_{ej}v_{ij}$$

where  $d$  is a hyperparameter, and  $\mathbf{U} \in \mathbb{R}^{\mathcal{E}^2 \times d}$  and  $\mathbf{V} \in \mathbb{R}^{\mathcal{J} \times d}$  are model parameters. The two other models are an inter-item neighborhood model and selectional preferences (described in Section 1.4.2.1), which we do not detail here. Training such a model is difficult since we do not have access to negative facts: not observing a sample  $(e, i) \notin \mathcal{D}_j$  does not necessarily imply that this sample is false. To cope with this issue, Riedel et al. (2013) propose to use the Bayesian personalized ranking model (BPR, Rendle et al. 2009). Instead of enforcing each element  $m_{ei}$  to be equal to 1 or 0, BPR relies upon a ranking objective pushing element observed to be true to be ranked higher than unobserved elements. This is done through a contrastive objective between observed positive samples and unobserved negative samples from a uniform distribution:

$$J_{\text{US}}(\theta) = \sum_{(e^+, i) \in \mathcal{D}_j} \sum_{\substack{(e^-, i) \in \mathcal{E}^2 \times \mathcal{J} \\ (e^-, i) \notin \mathcal{D}_j}} \log \sigma(m_{e^+i} - m_{e^-i})$$

This objective can be directly maximized using stochastic gradient ascent. Riedel et al. (2013) experiment on a NYT + FB dataset, this means the unsupervised dataset  $\mathcal{D}$  comes from the New York Times (NYT, Section C.5) and the knowledge base  $\mathcal{D}_{\text{KB}}$  is Freebase (FB, Section C.3).

Rendle et al., “BPR: Bayesian Personalized Ranking from Implicit Feedback”  
UAI 2009

#### 2.4.4 Aggregate PCNN Extraction

PCNN is a sentence-level feature extractor introduced in Section 2.3.6. Zeng et al. (2015) introduce the PCNN feature extractor together with a multi-instance learning algorithm. Given a bag of mentions  $\mathbf{x} \in \mathcal{D}^e$ , for each mention  $x_i \in \mathbf{x}$ , they model  $P(r | x_i; \theta)$ . However, the optimization is done over each bag of mentions separately:

Zeng et al., “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks”  
EMNLP 2015

$$\mathcal{L}_{\text{PCNN}}(\theta) = - \sum_{(\mathbf{x}, r) \in \mathcal{D}_{\mathcal{R}}^e} \log P(r | \mathbf{x}^*; \theta) \quad (2.6)$$

$$x^* = \operatorname{argmax}_{x_i \in \mathbf{x}} P(r | x_i; \theta) \quad (2.7)$$

In other words, for a set of mention  $\mathbf{x}$  of an entity pair, the network back-propagates only on the sample that predicts a relation with the highest certainty. Thus PCNN is a multi-instance single-relation model, it assumes  $\mathcal{H}_{\text{MULTI-INSTANCE}}$  but also  $\mathcal{H}_{\text{1-ADJACENCY}}$ .

Zeng et al. (2015) continue to use the experimental setup of Surdeanu et al. (2012), i.e. using a distantly supervised dataset, but complement it with a manual evaluation to have a better estimate of the precision.

Y. Lin et al. (2016) improve the PCNN model with an attention mechanism over mentions to replace the argmax of Equation 2.7. The attention mechanism’s memory is built from the output of the PCNN on each mention without applying a softmax; the PCNN is simply used to produce a representation for each mention. Equations 2.6 and 2.7 are then replaced

Y. Lin et al., “Neural Relation Extraction with Selective Attention over Instances”  
ACL 2016

3511 by:

3512

3513

$$\mathcal{L}_{\text{Lin}}(\boldsymbol{\theta}) = - \sum_{(\mathbf{x}, r) \in \mathcal{D}_x^e} \log P(r | \mathbf{x}; \boldsymbol{\theta})$$

3514

3515

$$P(r | \mathbf{x}; \boldsymbol{\theta}) \propto \exp(\mathbf{W}\mathbf{s}(\mathbf{x}, r) + \mathbf{b})$$

3516

3517

$$\mathbf{s}(\mathbf{x}, r) = \sum_{x_i \in \mathbf{x}} \alpha_i \text{PCNN}(x_i)$$

3518

3519 where the  $\alpha_i$  are attention weights computed from a bilinear product be-  
 3520 tween the query  $r$  and the memory  $\text{PCNN}(\mathbf{x})$ , similarly to the setup of  
 3521 Section 1.3.3. Y. Lin et al. (2016) show that this modification improves  
 3522 the results of PCNN, this can be seen as a relaxation of  $\mathcal{H}_{\text{MULTI-INSTANCE}}$ : the  
 3523 standard PCNN approach assumes that each fact in  $\mathcal{D}_{\text{KB}}$  is conveyed by  
 3524 a single sentence through its argmax; in contrast, the attention approach  
 3525 simply assumes that all facts are conveyed in  $\mathcal{D}$ , at least by one sentence  
 3526 but possibly by several ones.

3527

3528

3529

## 2.4.5 Entity Pair Graph

3530

3531 The multi-instance approach shares information at the entity pair level.  
 3532 However, information could also be shared between different entity pairs.  
 3533 This is the idea put forth by entity pair graph neural network (EPGNN,  
 3534 Zhao et al. 2019). The basic sharing unit becomes the entity: when two  
 3535 mentions  $(s, e_1, e_2), (s', e'_1, e'_2) \in \mathcal{D}$  share at least one entity ( $\{e_1, e_2\} \cap$   
 3536  $\{e'_1, e'_2\} \neq \emptyset$ ), their features interact with each other in order to make a  
 3537 prediction. The sharing of information is made following an entity pair  
 3538 graph that links together bags of mentions with a common entity as illus-  
 3539 trated in Figure 2.11.

3540

3541

3542

3543

3544

3545

3546

3547

3548

3549

3550

3551

3552

3553

3554

3555

3556

3557

3558

3559

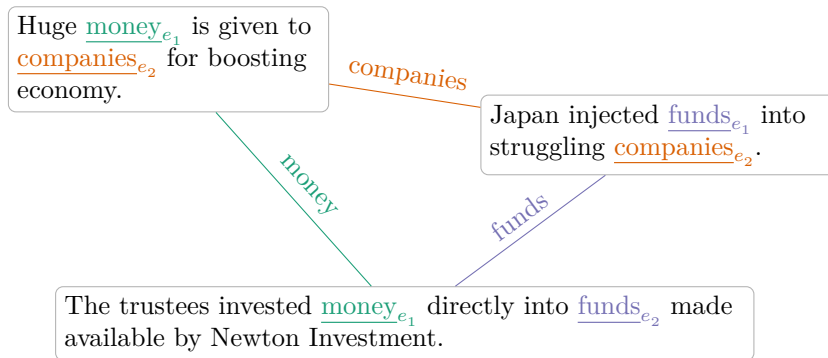
3560

3561

3562

3563

3564



Zhao et al., “Improving Relation Classification by Entity Pair Graph” PMLR 2019

Figure 2.11: Entity pair graph. Each node corresponds to a bag of mentions, each edge of the graph corresponds to an entity in common between the two bags, the edges are labeled with the shared entity. For illustration purpose, we show a single sample per bag. This example is from the SemEval 2010 Task 8 dataset (described in Section C.6). All sentences convey the *entity-destination* relation.

To obtain a distributed representation for a sentence, EPGNN uses BERT (Section 1.3.4). More precisely, it combines the embedding of the CLS token<sup>41</sup> with the embeddings corresponding to the two entities through a mean pooling. The sentence feature extraction architecture is illustrated by Figure 2.12. This is one of several methods to obtain an entity-aware fixed-size representation of a tagged sentence; other approaches are developed in Section 2.3.7.

Given a vector representation for each sentence in the dataset, we can label the vertices of the entity pair graph. A spectral graph convolutional network (GCN, Section 4.3.2) is then used to aggregate the information of its neighboring samples into each vertex. Thus, EPGNN produces two representations for a sample: one sentential and one topological. From

<sup>41</sup> As a reminder, the CLS token is the marker for the beginning of the sentence, its embedding purposes to represent the whole sentence.

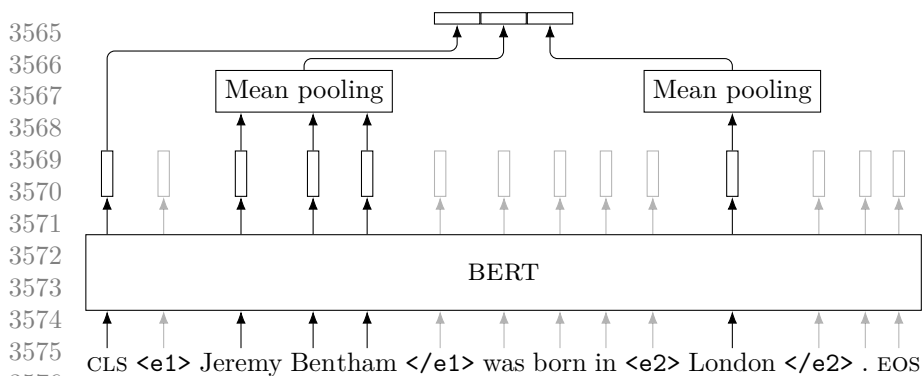


Figure 2.12: EPGNN sentence representation. “Bentham” was split into two subword tokens, “Ben-” and “-tham” by the BPE algorithm described in Section 1.2.3. The contextualized embeddings of most words are ignored. The final representation is only built using the entities span and the CLS token. Not appearing on the figure are linear layers used to post-process the output of the mean poolings and the final representation as well as a ReLU non-linearity. Compare to Figure 2.7.

these two representations, a prediction is made using a linear and softmax layer. Since a single relation is produced for each sample, EPGNN is trained using the usual classification cross-entropy loss. More details on graph-based approaches are given in Chapter 4.

Zhao et al. (2019) evaluate EPGNN on two datasets, SemEval 2010 Task 8 (Section C.6) and ACE 2005 (Section C.1). Reaching a half-directed macro- $\overline{F}_1$  of 90.2% on the first one, and a micro- $F_1$  of 77.1% on the second.

## 2.5 Unsupervised Extraction Models

In the unsupervised setting, no samples are labeled with a relation, i.e. all samples are triplets (sentence, head entity, tail entity) from  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$ . Furthermore, no information about the relation set  $\mathcal{R}$  is available. This is problematic since whether a specific semantic link is worthy of appearing in  $\mathcal{R}$  or not is not well defined. Having so little information about what constitutes a relation makes the problem intractable if we do not impose some restrictions upon  $\mathcal{R}$ . All unsupervised models presented in this section are not universal and make some kind of assumption on the structure of the data or on its underlying knowledge base. However, developing unsupervised relation extraction models is still interesting for three reasons: they (1) do not necessitate labeled data except for validating the models; (2) can uncover new relation types; and (3) can be trained from large unlabeled datasets and then fine-tuned for specific relations.

For all models, we list the important modeling hypothesis such as  $\mathcal{H}_{\text{1-ADJACENCY}}$  and  $\mathcal{H}_{\text{PULLBACK}}$  introduced previously. Appendix B contains a list of assumptions with some counterexamples and references to the sections where they were introduced. We strongly encourage the reader to refer to it, especially when the implications of a modeling hypothesis is not immediately clear.

### 2.5.1 Evaluation

The output of unsupervised models vary widely. The main modus operandi can be categorized into two categories:

**Clustering** A first approach is to cluster the samples such that all samples in the same cluster convey the same relation and samples in different clusters convey different relations.

“If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake.”

— Yann LeCun, Inaugural Lecture at Collège de France (2016)

3619 **Similarity Space** A second approach is to associate each sample with an  
 3620 element of a vector space equipped with a similarity function. If two  
 3621 samples are similar in this vector space, they convey similar relations.  
 3622 This can be seen as a soft version of the clustering approach.

3623 This distinction has an impact on how we evaluate the models. In the  
 3624 first case, standard clustering metrics are used. We introduce  $B^3$  (Bagga  
 3625 and Baldwin 1998), V-measure (Rosenberg and Hirschberg 2007) and ARI  
 3626 (Hubert and Arabie 1985) in Section 2.5.1.1. They are the most prevalent  
 3627 metrics in cluster evaluation,  $B^3$  in particular is widely used in unsuper-  
 3628 vised relation extraction. In the second case, a few-shot evaluation can be  
 3629 used (Han et al. 2018). We introduce this approach in Section 2.5.1.2.

3630 A difficulty of evaluating unlabeled clusters is that we do not know  
 3631 which cluster should be compared to which relation. A possible solution  
 3632 to this problem is to use a small number of labeled samples, which can be  
 3633 used to constrain the output of a model to fall into a specific relation set  $\mathcal{R}$ .  
 3634 This setup is actually similar to semi-supervised approaches such as label  
 3635 propagation (Section 2.4.1), except that the model must be trained in an  
 3636 unsupervised fashion before being fine-tuned on the supervised dataset.  
 3637 Similar to the label propagation model evaluation, unsupervised models  
 3638 evaluated by fine-tuning on a supervised dataset usually report perfor-  
 3639 mance varying the number of train labels. These performances are mea-  
 3640 sured using the standard supervised metrics introduced in Section 2.3.1.  
 3641 Evaluating performances as a pre-training method can be used for all un-  
 3642 supervised models, in particular similarity-space-based approaches.

### 3644 2.5.1.1 Clustering Metrics

3645 In this section, we describe three metrics used to evaluate clustering ap-  
 3646 proaches. The first metric,  $B^3$  was first introduced to unsupervised relation  
 3647 extraction by rel-LDA (Yao et al. 2011, Section 2.5.4), while the other two  
 3648 were proposed as complements by Simon et al. (2019) presented in Chap-  
 3649 ter 3.

3650 To clearly describe these different clustering metrics, we propose a  
 3651 common probabilistic formulation—in practice, these probabilities are es-  
 3652 timated on the validation and test sets—and use the following notations.  
 3653 Let  $X$  and  $Y$  be random variables corresponding to samples in the dataset.  
 3654 Following Section 2.3.1, we denote by  $c(X)$  the predicted cluster of  $X$  and  
 3655  $g(X)$  its conveyed gold relation.<sup>42</sup>

3656  $B^3$  The metric most commonly computed for unsupervised model eval-  
 3657 uation is a generalization of  $F_1$  for clustering tasks called  $B^3$  (Bagga and  
 3658 Baldwin 1998). The  $B^3$  precision and recall are defined as follows:

$$3662 \quad B^3 \text{ precision}(g, c) = \mathbb{E}_{X, Y \sim \mathcal{U}(\mathcal{D}_{\mathcal{R}})} P(g(X) = g(Y) \mid c(X) = c(Y))$$

$$3663 \quad B^3 \text{ recall}(g, c) = \mathbb{E}_{X, Y \sim \mathcal{U}(\mathcal{D}_{\mathcal{R}})} P(c(X) = c(Y) \mid g(X) = g(Y))$$

3664 As precision and recall can be trivially maximized by putting each sample  
 3665 in its own cluster or by clustering all samples into a single class, the main  
 3666 metric  $B^3 F_1$  is defined as the harmonic mean of precision and recall:

$$3667 \quad B^3 F_1(g, c) = \frac{2}{B^3 \text{ precision}(g, c)^{-1} + B^3 \text{ recall}(g, c)^{-1}}$$

““ *The cake is a lie.*  
 — Valve, “Portal” (2007)

<sup>42</sup> This implies that a labeled dataset is sadly necessary to evaluate an unsupervised clustering model.

Bagga and Baldwin, “Entity-Based Cross-Document Coreferencing Using the Vector Space Model” ACL 1998

3673 While the usual precision (Section 2.3.1) can be seen as the probability  
 3674 that a sample with a given prediction is correct, the  $B^3$  precision cannot  
 3675 use the correct relation as a reference to determine the correctness of a  
 3676 prediction. Instead, whether an assignment is correct is computed as the  
 3677 expectation that a sample is accurately classified relatively to all other  
 3678 samples grouped in the same cluster.

3679  
 3680 **V-measure** Another metric is the entropy-based V-measure (Rosenberg  
 3681 and Hirschberg 2007). This metric is defined by homogeneity and com-  
 3682 pleteness, which are akin to  $B^3$  precision and recall but rely on conditional  
 3683 entropy. For a cluster to be homogeneous, we want most of its elements to  
 3684 convey the same gold relation. In other words, the distribution of gold re-  
 3685 lations inside a cluster must have low entropy. This entropy is normalized  
 3686 by the unconditioned entropy of the gold relations to ensure that it does  
 3687 not depend on the size of the dataset:

$$3688 \text{homogeneity}(g, c) = 1 - \frac{H(c(X) | g(X))}{H(c(X))}.$$

3691 Similarly, for a cluster to be complete, we want all the elements conveying  
 3692 the same gold relation to be captured by this cluster. In other words, the  
 3693 distribution of clusters inside a gold relation must have low entropy:

$$3694 \text{completeness}(g, c) = 1 - \frac{H(g(X) | c(X))}{H(g(X))}.$$

3695  
 3696 As  $B^3$ , the V-measure is summarized by the  $F_1$  value:

$$3697 \text{V-measure}(g, c) = \frac{2}{\text{homogeneity}(g, c)^{-1} + \text{completeness}(g, c)^{-1}}.$$

3702 Compared to  $B^3$ , the V-measure penalizes small impurities in a relatively  
 3703 “pure” cluster more harshly than in less pure ones. Symmetrically,  
 3704 it penalizes a degradation of a well-clustered relation more than of a less-  
 3705 well-clustered one. This difference is illustrated in Figure 2.13.

3706  
 3707 **Adjusted Rand Index** The Rand index (RI, Rand 1971) is the last  
 3708 clustering metric we consider, it is defined as the probability that cluster  
 3709 and gold assignments are compatible:

$$3710 \text{RI}(g, c) = \mathbb{E}_{X,Y} [P(c(X) = c(Y) \Leftrightarrow g(X) = g(Y))]$$

3712 In other words, given two samples, the RI is improved when both samples  
 3713 are in the same cluster and convey the same gold relation or when both  
 3714 samples are in different clusters and convey different relations; otherwise,  
 3715 the RI deteriorates. The adjusted Rand index (ARI, Hubert and Arabie  
 3716 1985) is a normalization of the Rand index such that a random assignment  
 3717 has an ARI of 0, and the maximum is 1:

$$3718 \text{ARI}(g, c) = \frac{\text{RI}(g, c) - \mathbb{E}_{c \sim \mathcal{U}(\mathcal{R}^{\mathcal{D}})} [\text{RI}(g, c)]}{\max_{c \in \mathcal{R}^{\mathcal{D}}} \text{RI}(g, c) - \mathbb{E}_{c \sim \mathcal{U}(\mathcal{R}^{\mathcal{D}})} [\text{RI}(g, c)]}$$

3723 In practice, the ARI can be computed from the elements of the confusion  
 3724 matrix. Compared to the previous metrics, ARI will be less sensitive to a  
 3725 discrepancy between precision–homogeneity and recall–completeness since  
 3726 it is not a harmonic mean of both.

Rosenberg and Hirschberg, “V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure” EMNLP 2007

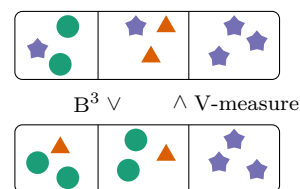


Figure 2.13: Comparison of  $B^3$  and V-measure. Samples conveying three different relations indicated by shape and color are clustered into three boxes. The two rows represent two different clusterings,  $B^3$  favors the first one while V-measure favors the second. V-measure prefers the second clustering since the blue star cluster is kept pure; on the other hand, the green circle cluster is impure no matter what, so its purity is not taken as much into account by the V-measure compared to  $B^3$ .

Hubert and Arabie, “Comparing partitions” JOC 1985

## 3727 2.5.1.2 Few-shot

3728 Clustering metrics are problematic since producing a clustering with no a  
3729 priori knowledge on the relation schema  $\mathcal{R}$  leads to unsolvable problems:  
3730

- 3731 • Should the relation *sibling* be cut into *brother* and *sister*?
- 3732
- 3733 • Is the relation between a country and its capital the same as the one  
3734 between a county and its seat?
- 3735
- 3736 • Is the ear *part of* the head in the same fashion that the star Altair  
3737 is *part of* the Aquila constellation?

3738 All of these questions can be answered differently depending on the de-  
3739 sign of the underlying knowledge base. However, unsupervised clustering  
3740 algorithms do not depend on  $\mathcal{R}$ . They must decide whether “Phaedra is  
3741 the sister of Ariadne” and “Castor is the brother of Pollux” go inside the  
3742 same cluster independently of these design choices.

3743 Fine-tuning on a supervised dataset solves this problem but adds an-  
3744 other. The evaluation no longer assesses the proficiency of a model to learn  
3745 from unlabeled data alone; it also evaluates its ability to adapt to labeled  
3746 samples. Furthermore, the smaller the labeled dataset is, the more results  
3747 have high variance. On the other hand, the larger the labeled dataset is,  
3748 the less the experiment evaluates the unsupervised phase.

3749 A few-shot evaluation can be used to answer these caveats. Instead  
3750 of evaluating a clustering of the samples, few-shot experiments evaluate  
3751 a similarity function between samples:  $\text{sim}: \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ . Given a query  
3752 sample  $x^{(q)}$  and a set of candidates  $\mathbf{x}^{(c)} = \{x_i^{(c)} \mid i = 1, \dots, C\}$ , the model  
3753 is evaluated on whether it is able to find the candidate conveying the same  
3754 relation as the query. This is simply reported as an accuracy by comparing  
3755  $\text{argmax}_{x \in \mathbf{x}^{(c)}} \text{sim}(x^{(q)}, x)$  with the correct candidate.  
3756

3757 **Query:**

3758 It flows into the Hörsel <sub>$e_2$</sub>  in Eisenach <sub>$e_1$</sub> .

3760 **Candidates:**

- 3761 It is remake of Hindi <sub>$e_2$</sub>  film “Tezaab <sub>$e_1$</sub> ”.
- 3762 Cynidr <sub>$e_1$</sub>  was the son of St Gwladys <sub>$e_2$</sub> .
- 3763 → Herron Island <sub>$e_1$</sub>  lies in Case Inlet <sub>$e_2$</sub> .
- 3764 He gained the support of Admiral <sub>$e_2$</sub>  Edward Russell <sub>$e_1$</sub> .
- 3765 NGC 271 <sub>$e_1$</sub>  is a spiral galaxy in the constellation Cetus <sub>$e_2$</sub> .

3768 Table 2.2 gives an example of a few-shot problem. It illustrates the  
3769 five-way one-shot problem, meaning that we must choose a relation among  
3770 five and that each of the five relations is represented by a single sample.  
3771 Another popular variant is the ten-way five-shot problem: the candidates  
3772 are split into ten bags of five samples each, all samples in a bag convey  
3773 the same relation, and the goal is to predict the bag in which the query  
3774 belongs. Candidates are sometimes referred to as “train set” and the query  
3775 as “test set” since this can be seen as an extremely small dataset with five  
3776 training samples and one test sample.

3777 FewRel, described in Section C.2, is the standard few-shot dataset. In  
3778 FewRel, Altair is not P361 *part of* Aquila, it is P59 *part of constellation*  
3779 Aquila. However, this design decision does not influence the evaluation.  
3780 Given the query “Altair is located in the Aquila constellation,” a model

This section only presents Few-shot evaluation. It is possible—and quite common—to train a model using a few-shot objective, usually as a fine-tuning phase before a few-shot evaluation. Since we are mostly interested in unsupervised approaches, we do not delve into few-shot training. See Han et al. (2018) for details.

$C$  is the number of candidates, in Table 2.2 we have  $C = 5$ .

Table 2.2: Few-shot problem. For ease of reading, the entity identifiers—such as Q450036 for “Hörsel”—are not given. Both the query and the third candidate convey the relation P206 *located in or next to body of water*.

Quite confusingly, they can also be referred to as “meta-train” and “meta-test.” Indeed, to follow the usual semantic of the “meta-” prefix, the “meta-sets” should refer to sets of (query, candidates) tuples, not the candidates themselves.



ought to rank this sample as more similar to samples conveying *part of constellation* than to those conveying other kinds of *part of* relationships. If FewRel made the opposite design choice, the model would still be able to achieve high accuracy by ensuring *part of* samples are similar. The decision to split or not the *part of* relation should be of no concern to the unsupervised model.

3787

3788

3789

## 2.5.2 Open Information Extraction

3790

3791

3792

3793

3794

3795

3796

3797

3798

3799

3800

3801

3802

3803

3804

3805

In Open information extraction (OIE, Banko et al. 2007), the closed-domain assumption (Section 2.1.1.2) is neither made for relations nor entities, which are extracted jointly. Instead  $\mathcal{E}$  and  $\mathcal{R}$  are implicitly defined from the language itself, typically a fact  $(e_1, r, e_2)$  is expressed as a triplet such as (noun phrase, verb phrase, noun phrase). This makes OIE particularly interesting when processing large amounts of data from the web, where there can be many unanticipated relations of interest.

This section focuses on TextRunner, the first model implementing OIE. It uses an aggregate extraction setup where  $\mathcal{D}$  is directly mapped to  $\mathcal{D}_{\text{KB}}$ , with the peculiarity that  $\mathcal{D}_{\text{KB}}$  is defined using surface forms only. The hypothesis on which TextRunner relies is that the surface form of the relation conveyed by a sentence appears in the path between the two entities in its dependency tree. In the OIE setup, these surface forms can then be used as labels for the conveyed relations, thereby using the language itself as the relation domain  $\mathcal{R}$ . TextRunner can be split into three parts:

3806

3807

3808

3809

3810

3811

3812

3813

3814

3815

3816

**The Learner** is a naive Bayes classifier, trained on a small dataset to predict whether a fact  $(e_1, r, e_2)$  is trustworthy. To extract a set of samples for this task, a dependency parser (Figure 2.4) is run on the dataset and tuples  $(e_1, r, e_2)$  are extracted where  $e_1$  and  $e_2$  are base noun phrases and  $r$  is the dependency path between the two entities. The tuples are then automatically labeled as trustworthy or not according to a set of heuristics such as the length of the dependency path and whether it crosses a sentence boundary. The naive Bayes classifier is then trained to predict the trustworthiness of a tuple given a set of hand-engineered features (Section 2.3.4).

3817

3818

3819

3820

3821

3822

3823

3824

3825

3826

3827

**The Extractor** extracts trustworthy facts on the whole dataset. The features on which the Learner is built only depend on part-of-speech (POS) tags (noun, verb, adjective...) such that the Extractor does not need to run a dependency parser on all the sentences in the entire dataset. While the Learner uses the dependency path for  $r$ , the Extractor uses the infix from which non-essential phrases (such as adverbs) are eliminated heuristically. Thus the Extractor simply runs a POS tagger on all sentences, finds all possible entities  $e$ , estimates a probable relation  $r$  and filters them using the Learner to output a set of trustworthy facts.

3828

3829

3830

3831

3832

3833

3834

**The Assessor** assigns a probability that a fact is true from redundancy in the dataset using the urns model of Downey et al. (2005). This model uses a binomial distribution to model the probability that a correct fact appears  $k$  times among  $n$  extractions with a fixed repetition rate. Furthermore, it assumes both correct and incorrect facts follow different Zipf's laws. The shape parameter  $s_I$  of the distribution of incorrect facts is assumed to be 1. While the shape parameter  $s_C$  of

Banko et al., "Open Information Extraction from the Web" IJCAI 2007

Dependency parsers tend to be a lot slower than POS taggers.

3835 the distribution of correct facts as well as the number of correct facts  
 3836  $N_C$  are estimated using an expectation–maximization algorithm. In  
 3837 the expectation step, the binomial and Zipf distribution assumptions  
 3838 can be combined using Bayes’ theorem to estimate whether a fact is  
 3839 correct or not. In the maximization step, the parameters  $s_C$  and  $N_C$   
 3840 are estimated.

3841 Banko et al. (2007) compare their approach to KnowItAll, an earlier  
 3842 work similar to OIE but needing a list of relations (surface forms) as input  
 3843 to define the target relation schema  $\mathcal{R}$ . On a set of ten relations, they  
 3844 manually labeled the extracted facts as correct or not, obtaining an error  
 3845 rate of 12% for TextRunner and 18% for KnowItAll. They further run  
 3846 their model on 9 million web pages, extracting 7.8 million facts.  
 3847

3848 A limitation of the OIE approach is that it heavily depends on the raw  
 3849 surface form and suffers from bad generalization. The two facts “Bletchley  
 3850 Park known as Station X” and “Bletchley Park codenamed Station X”  
 3851 are considered different by TextRunner since the surface forms conveying  
 3852 the relations in the underlying sentences are different. Subsequent OIE  
 3853 approaches try to address this problem, such as Yates et al. (2007), which  
 3854 extend TextRunner with a resolver (Yates and Etzioni 2007) to merge  
 3855 synonyms. However, this problem is not overcome yet and is still an active  
 3856 area of research. Furthermore, since the input of OIE systems is often taken  
 3857 to be the largest possible chunk of the web, and since the extracted facts  
 3858 do not follow a strict nomenclature, a fair evaluation of OIE systems among  
 3859 themselves or to other unsupervised relation extraction models is still not  
 3860 feasible.

3861

### 3862 2.5.3 Clustering Surface Forms

3863  
 3864 The first unsupervised relation extraction model was the clustering ap-  
 3865 proach of Hasegawa et al. (2004). It is somewhat similar to DIRT (Sec-  
 3866 tion 2.3.3) in that it uses a similarity between samples. However, their  
 3867 work goes one step further by using this similarity to build relation classes.  
 3868 Furthermore, Hasegawa et al. (2004) does not assume  $\mathcal{H}_{\text{PULLBACK}}$ , i.e. it does  
 3869 not assume that the sentence and entities convey the relation separately,  
 3870 on their own. Instead, its basic assumption is that the infix between two  
 3871 entities is the expression of the conveyed relation. As such, if two infixes  
 3872 are similar, the sentences convey similar relations. Furthermore, NER (see  
 3873 the introduction of Chapter 2) is performed on the text instead of sim-  
 3874 ple entity chunking. This means that all entities are tagged with a type  
 3875 such as “organization” and “person.” These types strongly constrain the  
 3876 relations through the following assumption:

3877

3878 **Assumption  $\mathcal{H}_{\text{TYPE}}$ :** *All entities have a unique type, and all relations are*  
 3879 *left and right restricted to one of these types.*

3880  $\exists \mathcal{T}$  partition of  $\mathcal{E} : \forall r \in \mathcal{R} : \exists X, Y \in \mathcal{T} : r \bullet \tilde{r} \cup \mathbf{1}_X = \mathbf{1}_X \wedge \tilde{r} \bullet r \cup \mathbf{1}_Y = \mathbf{1}_Y$   
 3881

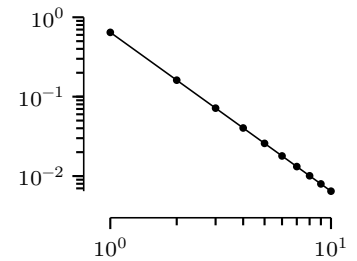
3882 This is a natural assumption for many relations; for example, the re-  
 3883 lation *born in* is always between a person and a geopolitical entity (GPE).

3884 Given a pair of entities  $(e_1, e_2) \in \mathcal{E}^2$ , Hasegawa et al. (2004) collect all  
 3885 samples in which they appear and extract a single vector representation  
 3886 from all these samples. This representation is built from the bag of words  
 3887 of the infixes weighted by TF–IDF (term frequency–inverse document fre-  
 3888 quency). Since a bag of words discards the ordering of the words or entities,

Zipf’s law comes from the externalist linguistic school. It follows from the ob-  
 servation that the frequency of the second most common word is half the one  
 of the most frequent word, that the one of the third most common word is a  
 third of the one of the most frequent, etc. The same distribution can often  
 be observed in information extraction. Zipf’s law is parametrized by a shape  
 $s$  and the number of elements  $N$ :

$$P(x | s) \propto \begin{cases} x^{-s} & \text{for } x \in \{1, \dots, N\} \\ 0 & \text{otherwise} \end{cases}$$

A Zipf’s law is easily recognizable on a log–log scale, its probability mass func-  
 tion being a straight line. Take for example the Zipf’s law with parameters  
 $s = 2$  and  $N = 10$ :



Hasegawa et al., “Discovering Relations among Named Entities from Large Corpora” ACL 2004

As a reminder, the infix is the span of text between the two entities in the sentence.

Following Section 1.4.1,  $\tilde{r}$  is the converse relation of  $r$ , i.e. the relation with  $e_1$  and  $e_2$  in the reverse order.  $\bullet$  is the composition operator and  $\mathbf{1}_X$  the complete relation over  $X$ .  $r \bullet \tilde{r}$  is the relation linking all the entities which appear as subject ( $e_1$ , on the left hand side) of  $r$  to themselves. This relation is constrained to be between entities in  $X$ . Less relevant to this formula,  $r \bullet \tilde{r}$  also links together entities linked by  $r$  to the same object.

Here, we assume that the partition  $\mathcal{T}$  is not degenerate and somewhat looks like a standard NER classification output. Otherwise,  $\mathcal{T} = \{\mathcal{E}\}$  is a valid partition of  $\mathcal{E}$ , and this assumption is tautological.

the variant of TF-IDF used takes into account the directionality:

$$\begin{aligned} \text{TF}(w, e_1, e_2) &= \text{number of times } w \text{ appears between } e_1 \text{ and } e_2 \\ &\quad - \text{number of times } w \text{ appears between } e_2 \text{ and } e_1 \end{aligned}$$

$$\text{IDF}(w) = (\text{number of documents in which } w \text{ appears})^{-1}$$

$$\text{TF-IDF}(w, e_1, e_2) = \text{TF}(w, e_1, e_2) \cdot \text{IDF}(w)$$

From this definition we obtain a representation  $\mathbf{z}_{e_1, e_2} \in \mathbb{R}^V$  of the pair  $(e_1, e_2) \in \mathcal{E}^2$  by taking the value of  $\text{TF-IDF}(w, e_1, e_2)$  for all  $w \in V$ . Given two entity pairs, their similarity is defined as follow:

$$\text{sim}(e, e') = \cos(\mathbf{z}_e, \mathbf{z}_{e'}) = \frac{\mathbf{z}_e \cdot \mathbf{z}_{e'}}{\|\mathbf{z}_e\| \|\mathbf{z}_{e'}\|}.$$

Using this similarity function, the complete-linkage clustering algorithm<sup>43</sup> (Defays 1977) is used to extract relations classes. Since each pair end up in a single cluster, this assumes  $\mathcal{H}_{1\text{-ADJACENCY}}$ . Hasegawa et al. (2004) evaluate their method on articles from the New York Times (NYT). They extract relations classes by first clustering all  $\mathbf{z}_{e_1, e_2}$  where  $e_1$  has the type person and  $e_2$  has the type GPE, and then by clustering all  $\mathbf{z}_{e_1, e_2}$  where both  $e_1$  and  $e_2$  are organizations. By clustering separately different type combinations, they ensure that  $\mathcal{H}_{\text{TYPE}}$  is enforced.

They furthermore experiment with automatic labeling of the clusters with the most frequent word appearing in the samples. Apart from the relation *prime minister*, which is simply labeled “minister” since only unigrams are considered, the labels are rather on point. To measure the performance of their model, they use a classical supervised  $F_1$  where each cluster is labeled by the majority gold relation. Using this somewhat unadapted metric, they reach an  $F_1$  of 82% on person-GPE pairs and an  $F_1$  of 77% on organization-organization pairs. This relatively high score compared to subsequent models can be explained by the small size of their dataset, which is further split by entity type. Furthermore, note that some generic relations such as *part of* do not follow  $\mathcal{H}_{\text{TYPE}}$  and, as such, cannot be captured.

#### 2.5.4 Rel-LDA

Rel-LDA (Yao et al. 2011) is a probabilistic generative model inspired by LDA. It works by clustering sentences: each relation defines a distribution over a handcrafted set of sentence features (Section 2.3.4) describing the relationship between the two entities in the text. Furthermore, rel-LDA models the propensity of a relation at the level of the document; thus, it is not strictly speaking a sentence-level relation extractor. The idea behind modeling this additional information is that when a relation such as *position played on team* appears in a document, other relations pertaining to sports are more likely to appear. Figure 2.14 gives the plate diagram for the rel-LDA model. It uses the following variables:

- $\mathbf{f}_i$  the features of the  $i$ -th sample, where  $f_{i,j}$  is its  $j$ -th feature
- $r_i$  the relation of the  $i$ -th sample
- $\theta_d$  the distribution of relations in the document  $d$
- $\phi_{r,j}$  the probability of the  $j$ -th feature to occurs for the relation  $r$
- $\alpha$  the Dirichlet prior for  $\theta_d$
- $\beta$  the Dirichlet prior for  $\phi_{r,j}$

<sup>43</sup> The complete-linkage algorithm is an agglomerative hierarchical clustering method also called farthest neighbor clustering. The algorithm starts with each sample in its own cluster then merges the clusters two by two until reaching the desired number of clusters. At each step, the two closest clusters are merged together, with the distance between clusters being defined as the distance between their farthest elements.

3943 The generative process is listed as Algorithm 2.4. The learning process  
 3944 uses the expectation–maximization algorithm. In the variational E-step,  
 3945 the relation for each sample  $r_i$  is sampled from the categorical distribution:

3946

3947

3948

3949

$$P(r_i | \mathbf{f}_i, d) \propto P(r_i | d) \prod_{j=1}^m P(f_{ij} | r_i)$$

3950 where  $P(r | d)$  is defined by  $\theta_d$  and  $P(f_{ij} | r)$  is defined by  $\phi_{rj}$ . In the  
 3951 M-step, the values for  $\theta_d$  are computed by counting the number of times  
 3952 each relation appears in  $d$  and the hyperprior  $\alpha$ ; and the value for  $\phi_{rj}$  is  
 3953 computed from the number of co-occurrences of the  $j$ -th feature with the  
 3954 relation  $r$  and from  $\beta$ .

3955 Yao et al. (2011) evaluate their model on the New York Times by  
 3956 comparing their clusters to relations in Freebase. However, because of the  
 3957 incompleteness of knowledge bases, they only evaluate the recall on Free-  
 3958 base and use manual annotation to estimate the precision. Even though  
 3959 the original article lacks a significant comparison, subsequent approaches  
 3960 often compare to rel-LDA.

3961 A first limitation of their approach is that given the relation  $r$ , the  
 3962 features  $f$  are independents. Since the entities are among those features,  
 3963 this means that  $P(e_2 | e_1, r) = P(e_2 | r)$  which is clearly false.

3964

3965 **Assumption  $\mathcal{H}_{\text{BICLIQUE}}$ :** *Given a relation, the entities are independent of*  
 3966 *one another:  $e_1 \perp e_2 | r$ . In other words, given a relation, all possible head*  
 3967 *entities are connected to all possible tail entities.*

$$3968 \forall r \in \mathcal{R} : \exists A, B \subseteq \mathcal{E} : r \bullet \tilde{r} = \mathbf{1}_A \wedge \tilde{r} \bullet r = \mathbf{1}_B$$

3969

3970 This is a widespread problem with generative models which are in-  
 3971 clined to make extensive independence assumptions. Furthermore, gener-  
 3972 ative models have an implicit bias that all observed features are related to  
 3973 relation extraction, even though they might measure other aspect of the  
 3974 sample (style, idiolectal word choice, etc). This might result in the model  
 3975 focusing on features not related to the relation extraction task.

3976 Several extensions of rel-LDA were proposed. Type-LDA (Yao et al.  
 3977 2011) purpose to model entity types which are latent variables of entity  
 3978 features, themselves generated from the relation variable  $r$ , thus softly en-  
 3979 forcing  $\mathcal{H}_{\text{TYPE}}$ . Sense-LDA (Yao et al. 2012) use a LDA-like model for each  
 3980 different dependency path. Clusters for different paths are then merged  
 3981 into relation clusters.

3982 Rel-LDA is an important work in that it proposes a simple evaluation  
 3983 framework; in particular, it introduces the  $B^3$  metric to unsupervised re-  
 3984 lation extraction. However, it predates the advent of neural networks and  
 3985 distributed representations in relation extraction, by which it was bound  
 3986 to be replaced.

3987

3988

3989

3990

3991

3992

3993

3994

3995

3996

3997

3998

3999

4000

## 2.5.5 Variational Autoencoder for Relation Extraction

Marcheggiani and Titov (2016) were first to propose a discriminative un-  
 supervised relation extraction model. Discriminative models directly solve  
 the inference problem of finding the posterior  $P(r | x)$ . This is in con-  
 trast to generative models such as rel-LDA which determine  $P(x | r)$  and  
 then use Bayes' theorem to compute  $P(r | x)$  and make a prediction. The  
 model of Marcheggiani and Titov (2016) is closely related to the approach  
 presented in Chapter 3. It is a clustering model, meaning that it produces

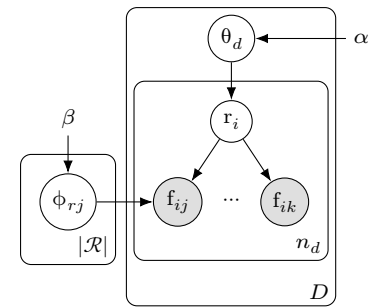


Figure 2.14: Rel-LDA plate diagram.  $D$  is the number of documents in the dataset and  $n_d$  is the number of samples in the document  $d$ . For each sample  $i$ , there are several features  $f_{i1}, f_{i2}, \dots, f_{im}$ , accordingly for each relation  $r$ , there are also several feature priors  $\phi_{r1}, \dots, \phi_{rm}$ , however for simplicity, a single prior is shown here.

**algorithm** REL-LDA GENERATION

Inputs:  $\alpha$  relations hyperprior

$\beta$  features hyperprior

Output:  $\mathbf{F}$  observed features

**for all** relations  $r$  **do**

**for all** features  $j$  **do**

    Choose  $\phi_{rj} \sim \text{Dir}(\beta)$

**for all** documents  $d$  **do**

    Choose  $\theta_d \sim \text{Dir}(\alpha)$

**for all** samples  $i$  in  $d$  **do**

      Choose  $r \sim \text{Cat}(\theta_d)$

**for all** features  $j$  **do**

        Choose  $f_{ij} \sim \text{Cat}(\phi_{rj})$

**output**  $\mathbf{F}$

Algorithm 2.4: The rel-LDA generative process. Dir are Dirichlet distributions. Cat are categorical distributions.

Yao et al., “Unsupervised Relation Discovery with Sense Disambiguation” ACL 2012

Marcheggiani and Titov, “Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations” TACL 2016

3997 clusters of samples where the samples in each cluster all convey the same  
 3998 relation. To do so, it uses a variational autoencoder model (VAE, Kingma  
 3999 and Welling 2014) that we now describe.

4000  
 4001 **Variational Autoencoder** The goal of a variational autoencoder is to  
 4002 learn a latent variable  $\mathbf{z}$  which explains the distribution of an observed  
 4003 variable  $\mathbf{x}$ . For our problem, the latent variable corresponds to the relation  
 4004 conveyed by the sample  $\mathbf{x}$ . We assume we know the generative process  
 4005  $P(\mathbf{x} | \mathbf{z}; \boldsymbol{\theta})$ , i.e. this process is the “decoder” (parametrized by  $\boldsymbol{\theta}$ ): given  
 4006 the latent variable it produces a sample. However, the process of interest  
 4007 to us is to estimate the latent variable—the relation—from a sample, that  
 4008 is  $P(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta})$ . Using Bayes’ theorem we can reformulate this posterior as  
 4009  $P(\mathbf{x} | \mathbf{z}; \boldsymbol{\theta})P(\mathbf{z} | \boldsymbol{\theta}) / P(\mathbf{x} | \boldsymbol{\theta})$ . However, computing  $P(\mathbf{x} | \boldsymbol{\theta})$  is often  
 4010 intractable, especially when the likelihood  $P(\mathbf{x} | \mathbf{z}; \boldsymbol{\theta})$  is modeled using  
 4011 a complicated function like a neural network. To solve this problem, a  
 4012 variational approach is used: another model  $Q$  parametrized by  $\phi$  is used  
 4013 to approximate  $P(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta})$  as well as possible. This approximation  $Q(\mathbf{z} |$   
 4014  $\mathbf{x}; \phi)$  is the “encoder” since it finds the latent variable associated with a  
 4015 sample. The model can then be trained by maximizing the log-likelihood  
 4016 given the latent variable estimated by  $Q$  and by minimizing the difference  
 4017 between the latent variable predicted by  $Q$  and the desired prior  $P(\mathbf{z} | \boldsymbol{\theta})$ :

$$4019 J_{\text{ELBO}}(\boldsymbol{\theta}, \phi) = \mathbb{E}_{Q(\mathbf{z}|\mathbf{x};\phi)} [\log P(\mathbf{x} | \mathbf{z}; \boldsymbol{\theta})] - D_{\text{KL}}(Q(\mathbf{z} | \mathbf{x}; \phi) \| P(\mathbf{z} | \boldsymbol{\theta})) \quad (2.8)$$

4021 A justification for this objective can also be found in the fact that it’s a  
 4022 lower bound of the log marginal likelihood  $\log P(\mathbf{x} | \boldsymbol{\theta})$ , hence its name:  
 4023 evidence lower bound (ELBO). The first part of the objective is often re-  
 4024 ferred to as the negative reconstruction loss since it seeks to reconstruct  
 4025 the sample  $\mathbf{x}$  after it went through the encoder  $Q$  and the decoder  $P$ . One  
 4026 last problem with the VAE approximation relates to the reconstruction  
 4027 loss, the estimation of the expectation over  $Q(\mathbf{z} | \mathbf{x}; \phi)$  not being differ-  
 4028 entiable which makes the model—in particular  $\phi$ —untrainable by gradient  
 4029 descent. This is usually solved using the reparameterization trick: sam-  
 4030 pling from  $Q(\mathbf{z} | \mathbf{x}; \phi)$  can often be done in a two steps process: sampling  
 4031 from a simple distribution like  $\epsilon \sim \mathcal{N}(0, 1)$  then transforming this sample  
 4032 using a deterministic process parametrized by  $\phi$ . The plate diagram of the  
 4033 VAE is given Figure 2.15 where the model  $P$  is marked with solid lines and  
 4034 the variational approximation  $Q$  is marked with dashed lines.

4036 Coming back to the model of Marcheggiani and Titov (2016), it is a  
 4037 conditional  $\beta$ -VAE,<sup>44</sup> i.e. the whole process is conditioned on an additional  
 4038 variable. Indeed, in their approach, only the entities  $\mathbf{e} \in \mathcal{E}^2$  are recon-  
 4039 structed, while the sentence  $s \in \mathcal{S}$  simply conditions the whole process.  
 4040 The latent variable explaining the observed entities is expected to be the  
 4041 relation conveyed by the sample. The resulting model’s plate diagram is  
 4042 given in Figure 2.16. This approach is defined by two models:

4044 **The Encoder**  $Q(r | e, s; \phi)$  is the relation extraction model properly  
 4045 speaking. It is defined as a linear model on top of handcrafted fea-  
 4046 tures (Section 2.3.4). For each sample, the model outputs a distri-  
 4047 bution over a predefined number of relations.

4049 **The Decoder**  $P(e | r; \boldsymbol{\theta})$  is a model estimating how likely it is for two  
 4050 entities to be linked by a relation. It is a reconstruction model since

Kingma and Welling, “Auto-Encoding Variational Bayes” ICLR 2014

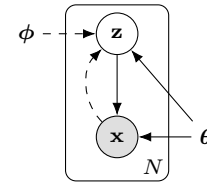


Figure 2.15: VAE plate diagram.  $N$  is the number of samples in the dataset.

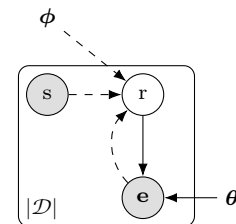


Figure 2.16: Marcheggiani and Titov (2016) plate diagram.

<sup>44</sup> The  $\beta$  in “ $\beta$ -VAE” simply indicates that the Kullback–Leibler term in Equation 2.8 is weighted by a hyperparameter  $\beta$ . More details are given in Chapter 3.

4051 the entities  $e$  are known and need to be retrieved from the latent  
 4052 relation  $r$  sampled from the encoder. It is defined using selectional  
 4053 preferences (Section 1.4.2.1) and RESCAL (Section 1.4.2.2).

4054 Note that to label a sample  $(e, s) \in \mathcal{D}$ , Marcheggiani and Titov (2016)  
 4055 simply select  $\operatorname{argmax}_{r \in \mathcal{R}} Q(r \mid e, s; \phi)$ , meaning that the decoder is not  
 4056 used during evaluation. Its sole purpose is to provide a supervision signal  
 4057 to the encoder through the maximization of  $J_{\text{ELBO}}$ . The whole autoencoder  
 4058 can also be interpreted as being trained by a surrogate task of filling-in  
 4059 entity blanks. This is the interpretation we use in Chapter 3.

4061 For Equation 2.8 to be well defined, a prior on the relations must also be  
 4062 selected; Marcheggiani and Titov (2016) make the following assumption:

4063 **Assumption  $\mathcal{H}_{\text{UNIFORM}}$ :** *All relations occur with equal frequency.*

$$4065 \forall r \in \mathcal{R}: P(r) = \frac{1}{|\mathcal{R}|}$$

4067 They evaluate their approach on the New York Times distantly super-  
 4068 vised by Freebase. By inducing 100 clusters, they show an improvement of  
 4069 the  $B^3 F_1$  compared to DIRT (Section 2.3.3) and rel-LDA (Section 2.5.4).  
 4070 They also experiment using semi-supervised evaluation (Section 2.5.1) by  
 4071 pre-training their decoder on a subset of Freebase before training their  
 4072 encoder as described above; this additional supervision improves the  $F_1$   
 4073 by more than 27%. These results were further improved by Yuan and El-  
 4074 dardiry (2021), which proposed to split the latent variable into a relation  
 4075  $r$  and sentence information  $z$ , with  $z$  conditioned on  $r$  and using a loss  
 4076 including the reconstruction of the sentence  $s$  from  $z$ .

## 4079 2.5.6 Matching the Blanks

4081 Matching the blanks (MTB, Soares et al. 2019) is an unsupervised method  
 4082 that does not attempt to cluster samples but rather learns a represen-  
 4083 tation of the relational semantics they convey. More precisely, this rep-  
 4084 resentation is used to measure the similarity between samples such that  
 4085 similar samples convey similar relations. As such, it is either evaluated  
 4086 as a supervised pre-training method (Section 2.5.1) or using a few-shot  
 4087 dataset (Section 2.5.1.2). The MTB article introduces several methods to  
 4088 extract an entity-aware representation of a sentence using BERT; this was  
 4089 discussed in Section 2.3.7. This section focuses on the unsupervised train-  
 4090 ing. As a reminder, we refer to sentence encoder of MTB by the function  
 4091  $\text{BERTcoder}: \mathcal{S} \rightarrow \mathbb{R}^d$  illustrated Figure 2.7. Given this encoder, MTB defines  
 4092 the similarity between samples as:

$$4094 \operatorname{sim}(s, s') = \sigma(\text{BERTcoder}(s)^\top \text{BERTcoder}(s')) \quad (2.9)$$

4096 This similarity function can be used to evaluate the model on a few-  
 4097 shot task. Note that this function completely ignores entities identifiers  
 4098 (e.g. Q211539), but can still exploit the entities surface forms (e.g. “Peter  
 4099 Singer”) through the sentence  $s \in \mathcal{S}$ . This model can be used as is, without  
 4100 any training other than the masked language model pre-training of BERT  
 4101 (Section 1.3.4.2) and reach an accuracy of 72.9% on the FewRel 5 way 1  
 4102 shot dataset.

4103 Soares et al. (2019) propose a training objective to fine-tune BERT for  
 4104 the unsupervised relation extraction task. This objective is called matching

Soares et al., “Matching the Blanks: Distributional Similarity for Relation Learning” ACL 2019

4105 the blanks. It assumes that two sentences containing the same entities convey  
 4106 the same relation. This is exactly  $\mathcal{H}_{1\text{-ADJACENCY}}$  as given Section 2.3.2.  
 4107 The probability that two sentences convey the same relation ( $D = 1$ ) is  
 4108 taken from the similarity function:  $P(D = 1 | s, s') = \text{sim}(s, s')$ . Given  
 4109 this, the  $\mathcal{H}_{1\text{-ADJACENCY}}$  assumption is translated into the following negative  
 4110 sampling (Section 1.2.1.3) loss:

$$4111 \mathcal{L}_{\text{MTB}} = \frac{-1}{|\mathcal{D}|^2} \sum_{\substack{(e,s) \in \mathcal{D} \\ (e',s') \in \mathcal{D}}} \delta_{e,e'} \log P(D = 1 | s, s') \quad (2.10)$$

$$4112 \quad \quad \quad + (1 - \delta_{e,e'}) \log P(D = 0 | s, s')$$

4116 This loss is minimized through gradient descent by sampling random posi-  
 4117 tive and negative sentence pairs. These pairs can be obtained by comparing  
 4118 the entity identifier without the need for any supervision.

4119 A problem with this approach is that the BERTcoder model can simply  
 4120 learn to perform entity linking on the entities surface forms in the sentences  
 4121  $s$ , thus minimizing Equation 2.10 by predicting whether  $e = e'$ . We want  
 4122 to avoid this since this would only work on samples seen during training  
 4123 and would not generalize to unseen entities. To ensure the model predicts  
 4124 whether the samples convey the same relation from the sentences  $s$  and  $s'$   
 4125 alone, blanks are introduced. A special token  $\langle \text{BLANK} \rangle$  is substituted to  
 4126 the entities as follow:

4127  $\langle \text{BLANK} \rangle_{e_1}$ , inspired by Cale’s earlier cover, recorded one of  
 4128 the most acclaimed versions of “ $\langle \text{BLANK} \rangle_{e_2}$ .”  
 4129  $\langle \text{BLANK} \rangle_{e_1}$ ’s rendition of “ $\langle \text{BLANK} \rangle_{e_2}$ ” has been called  
 4130 “one of the great songs” by Time...

4132 This is similar to the sample corruption of BERT (Section 1.3.4.2), indeed  
 4133 like BERT, the entity surface forms are blanked only a fraction<sup>45</sup> of the time  
 4134 so as to not confuse the model when real entities appear during evaluation.

4136 Another problem with Equation 2.10 is that the negative sample space  
 4137  $e \neq e'$  is extremely large. Instead of taking negative samples randomly  
 4138 in this space, Soares et al. (2019) propose to take only samples which are  
 4139 likely to be close to positive ones. To this end, the  $e \neq e'$  condition is  
 4140 actually replaced with the following one:

$$4141 |\{e_1, e_2\} \cap \{e'_1, e'_2\}| = 1$$

4143 These are called “strong negatives”: negative samples that have precisely  
 4144 one entity in common. Negative sampling, especially with strong negatives,  
 4145 leads to another unfortunate assumption:

4147 **Assumption  $\mathcal{H}_{1 \rightarrow 1}$ :** *All relations are one-to-one.*

$$4148 \forall r \in \mathcal{R}: r \bullet \tilde{r} \cup \mathbf{I} = \tilde{r} \bullet r \cup \mathbf{I} = \mathbf{I}$$

4150 Indeed, if a relation is not one-to-one, then there exists two facts  $e_1 r e_2$   
 4151 and  $e_1 r e_3$  (or respectively with  $\tilde{r}$ ); however these two facts form a strong  
 4152 negative pair, therefore as per  $\mathcal{L}_{\text{MTB}}$  their representations must be pulled  
 4153 away from one another.

4154 Despite these assumptions, MTB showcase impressive results, both as  
 4155 a few-shot and supervised pre-training method. It obtained state-of-the-  
 4156 art results both on the SemEval 2010 Task 8 dataset with a macro- $\overline{F}_1$  of  
 4157 82.7% and on FewRel with an accuracy of 90.1% on the 5 way 1 shot task.

45 Soares et al. (2019) blanks each entity with a probability of 70%, meaning that only 9% of training samples have both of their entity surface forms intact.

As a reminder,  $\overline{F}_1$  is the half-directed metric described Section 2.3.1. It is referred to as “taking directionality into account” in the SemEval dataset.

### 4159 2.5.7 SelfORE

4160 SelfORE (X. Hu et al. 2020) is a clustering approach similar to the one of  
 4161 Hasegawa et al. (2004) presented in Section 2.5.3 but using deep neural  
 4162 network models for extracting sentence representations and for grouping  
 4163 these representations into relation clusters. Since they follow the experi-  
 4164 mental setup of Simon et al. (2019), which we present in Chapter 3, their  
 4165 results are listed in that chapter.

4167 SelfORE uses MTB’s entity markers–entity start BERTcoder sentence  
 4168 representation. A clustering algorithm could be run to produce relation  
 4169 classes from these representations a la Hasegawa et al. (2004). However, X.  
 4170 Hu et al. (2020) introduce an iterative scheme to purify the clusters. This  
 4171 scheme is illustrated in Figure 2.17 and works by alternatively optimizing  
 4172 two losses  $\mathcal{L}_{AC}$  and  $\mathcal{L}_{RC}$ .

4173 The first loss  $\mathcal{L}_{AC}$  is the clustering loss which comes from DEC (Xie  
 4174 et al. 2016). DEC is a deep clustering algorithm that uses a denoising au-  
 4175 toencoder (Vincent et al. 2010) to compress the input. In their case, the  
 4176 input  $\mathbf{h}$  is the sentence encoded by BERTcoder. The denoising autoencoder  
 4177 is trained layer by layer with a small bottleneck which produces a com-  
 4178 pressed representation of the sentence  $\mathbf{z} = \text{Encoder}(\mathbf{h})$ . This is the space  
 4179 in which the clustering occurs. For each cluster  $j = 1, \dots, K$ , a centroid<sup>46</sup>  
 4180  $\boldsymbol{\mu}_j$  is learned such that a sentence is part of the cluster whose centroid  
 4181 is the closest to its compressed representation. This is modeled with a  
 4182 Student’s  $t$ -distribution with one degree of freedom centered around the  
 4183 centroid:

$$4184 q_{ij} = \frac{(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_j\|^2)^{-1}}{\sum_k (1 + \|\mathbf{z}_i - \boldsymbol{\mu}_k\|^2)^{-1}}$$

4187 To force the initial clusters to be more distinct, a target distribution  $p$   
 4188 is defined as:

$$4189 p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_k q_{ik}^2 / f_k} \quad (2.11)$$

4191 where  $f_j = \sum_i q_{ij}$  are soft cluster frequencies. To push  $\mathbf{Q}$  towards  $\mathbf{P}$ , a  
 4192 Kullback–Leibler divergence is used:

$$4195 \mathcal{L}_{AC} = D_{KL}(\mathbf{P} \parallel \mathbf{Q}) = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^K p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

4198 This loss is minimized by backpropagating to the cluster centroids  $\boldsymbol{\mu}_j$  and  
 4199 to the encoder’s parameters in the DAE. Note that the decoder of the  
 4200 DAE is only used for initializing the encoder such that the input can be  
 4201 reconstructed.

4202 Optimizing  $\mathcal{L}_{AC}$  is the first step of SelfORE; it assigns a pseudo-label  
 4203 to each sample in the dataset. The second step is to train a classifier  
 4204 to predict these pseudo-labels. The classifier is a simple multi-layer per-  
 4205 ceptron trained with the usual cross-entropy classification loss, which is  
 4206 called  $\mathcal{L}_{RC}$  in SelfORE. This loss also backpropagate to the BERTcoder  
 4207 thus changing the sentence representations  $\mathbf{h}$ . SelfORE is an iterative algo-  
 4208 rithm: changing the  $\mathbf{h}$  modifies the clustering found by DEC. Thus, the two  
 4209 steps, clustering and classification, are repeated several times until a stable  
 4210 label assignment is found.

4211 The central assumption of SelfORE is that BERTcoder already produces  
 4212 a good representation for relation extraction, which, as we saw with the

X. Hu et al., “SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction” EMNLP 2020

Xie et al., “Unsupervised Deep Embedding for Clustering Analysis” ICML 2016

<sup>46</sup> The  $k$ -means clustering algorithm is used to initialize the centroids. In practice, the  $k$ -means clusters could directly be used as soft labels. However, X. Hu et al. (2020) show that this underperforms compared to refining the clusters with  $\mathcal{L}_{AC}$ .

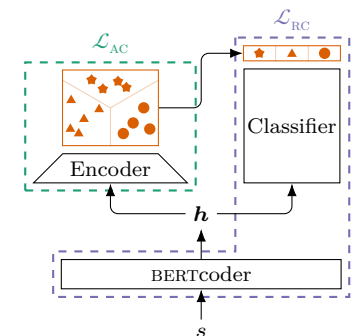


Figure 2.17: SelfORE iterative algorithm.



4213 non-fine-tuned BERTcoder score on FewRel in Section 2.5.6, is rather accu-  
 4214 rate. However, SelfORE also assumes  $\mathcal{H}_{\text{UNIFORM}}$ , i.e. that all relations appear  
 4215 with the same frequency. This assumption is enforced by  $\mathcal{L}_{\text{AC}}$ , through the  
 4216 normalization of the target distribution  $\mathbf{P}$  by soft cluster frequencies  $f_j$ .<sup>47</sup>  
 4217 Indeed, the distribution  $\mathbf{P}$  is the original distribution  $\mathbf{Q}$  more concentrated  
 4218 (because of the square) and more uniform (because of the normalization  
 4219 by  $f_j$ ).

4220 The interpretation of the concentration effect in terms of modeling  
 4221 hypotheses is more complex. The variable  $\mathbf{h}$  is the concatenation of the  
 4222 two entity embeddings. Let’s break down the BERTcoder function into two  
 4223 components:  $\text{ctx}_1(s)$  and  $\text{ctx}_2(s)$ . These are simply the two contextualized  
 4224 embeddings of  $\langle \mathbf{e}_1 \rangle$  and  $\langle \mathbf{e}_2 \rangle$  (Section 2.5.6), in other words the function  
 4225  $\text{ctx}$  contextualize an entity surface form inside its sentence. When two  
 4226 sentence representations  $\mathbf{h}$  and  $\mathbf{h}'$  are close, their pseudo-labels tend to be  
 4227 the same, and thus their relation also tend to be the same. In other words:  
 4228

4229 **Assumption  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$ :** *Two samples with the same contextualized*  
 4230 *representation of their entities’ surface forms convey the same relation.*  
 4231

4232  $\forall (s, \mathbf{e}, r), (s', \mathbf{e}', r') \in \mathcal{D}_{\mathcal{R}}$ :  
 4233 
$$\text{ctx}_1(s) = \text{ctx}_1(s') \wedge \text{ctx}_2(s) = \text{ctx}_2(s') \implies r = r'$$

4234 If we assume BERTcoder only performs entity linking of the entities  
 4235 surface form, then  $\text{ctx}_i(s) = e_i$  for  $i = 1, 2$ , in this case  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$   
 4236 collapses to  $\mathcal{H}_{1\text{-ADJACENCY}}$ , the contextualization inside the sentence  $s$  is  
 4237 ignored. On the other hand, if we assume BERTcoder provides no infor-  
 4238 mation about the entities and only encode the sentence, then  $\text{ctx}_i(s) = s$   
 4239 for  $i = 1, 2$  and  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$  only states that the entity identifiers  
 4240  $\mathbf{e} \in \mathcal{E}^2$  should have no influence on the relation. The effective repercussion  
 4241 of  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$  lies somewhere half-way between these two extremes.  
 4242

## 4243 2.6 Conclusion

4244 In this chapter, we introduced the relation extraction tasks (Section 2.1)  
 4245 and the different supervision schema with which we can tackle them (Sec-  
 4246 tion 2.2). As we showed, the development of supervised relation extrac-  
 4247 tion models closely followed the evolution of NLP models introduced in  
 4248 Section 1.3. This is particularly visible in Section 2.3, which follows the  
 4249 progress of sentential relation extraction approaches. Furthermore, the ex-  
 4250 pansion of the scale at which problems are tackled is visible both on the  
 4251 NLP side with the word-level to sentence-level evolution and on the infor-  
 4252 mation extraction side with the sentential to aggregate extraction evolu-  
 4253 tion. The aggregate models, which are more aligned with the information  
 4254 extraction field, are presented in Section 2.4. Within these models, we also  
 4255 see the evolution from the simple max-pooling of MIML (Section 2.4.2)  
 4256 toward more sophisticated approaches which model the topology of the  
 4257 dataset more finely (Section 2.4.5).  
 4258

4259 We limited our presentation of supervised models to those critical to  
 4260 the development of unsupervised models. Several recent approaches pro-  
 4261 pose to reframe supervised relation extraction—and other tasks—as lan-  
 4262 guage modeling (Raffel et al. 2020) or question answering (Cohen et al.  
 4263 2021) tasks. Since these approaches were not explored in the unsupervised  
 4264 setup yet, we omit them from our related work.  
 4265

<sup>47</sup> For further details, Xie et al. (2016) contains an analysis of the DEC clustering algorithm on imbalanced MNIST data.

Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer” JMLR 2020

Cohen et al., “Relation Classification as Two-way Span-Prediction” *under review* 2021

4267 Finally, Section 2.5 focused on the specific setup of interest to this the-  
4268 sis: unsupervised relation extraction. This setup is particularly complex  
4269 due to the discrepancy between the expressiveness of our supervised mod-  
4270 els and the weakness of the semantic signal we are seeking to extract. As  
4271 we saw, modeling hypotheses are central to tackling this problem. Early  
4272 models, including supervised ones, relied on strong hypotheses to facilitate  
4273 training. However, while supervised models can now use deep neural net-  
4274 works without any hypothesis other than the unbiasedness of their data,  
4275 unsupervised models still need to rely on strong assumptions.

4276 In the next section, we focus on unsupervised discriminative models,  
4277 in particular the VAE model presented in Section 2.5.5. In particular, we  
4278 propose better losses for enforcing  $\mathcal{H}_{\text{UNIFORM}}$ , which avoid problematic de-  
4279 generate solutions of the clustering relation extraction task.

4280

4281

4282

4283

4284

4285

4286

4287

4288

4289

4290

4291

4292

4293

4294

4295

4296

4297

4298

4299

4300

4301

4302

4303

4304

4305

4306

4307

4308

4309

4310

4311

4312

4313

4314

4315

4316

4317

4318

4319

4320

## Chapter 3

# Regularizing Discriminative Unsupervised Relation Extraction Models

All the works presented thus far follow the same underlying dynamic. There is a movement away from symbolic representations toward distributed ones, as well as a movement away from shallow models toward deeper ones. This can be seen in word, sentence and knowledge base representations (Chapter 1), as well as in relation extraction (Chapter 2). As we exposed in Chapter 2, a considerable amount of work has been conducted on supervised or weakly-supervised relation extraction (Sections 2.3 and 2.4), with recent state-of-the-art models using deep neural networks (Section 2.3.6). However, human annotation of text with knowledge base triplets is expensive and virtually impractical when the number of relations is large. Weakly-supervised methods such as distant supervision (Section 2.2.2) are also restricted to a handcrafted relation domain. Going further, purely unsupervised relation extraction methods working on raw texts, without any access to a knowledge base, have been developed (Section 2.5).

The first unsupervised models used a clustering (Section 2.5.3) or generative (Section 2.5.4) approach. The latter, which obtained state-of-the-art performance, still makes a lot of simplifying hypotheses, such as  $\mathcal{R}_{\text{BI-CLIQUE}}$ , assuming that the entities are conditionally independent between themselves given the relation. We posit that discriminative approaches can help further expressiveness, especially considering recent results with neural network models. The open question then becomes how to provide a sufficient learning signal to the classifier. The VAE model of Marcheggiani and Titov (2016) introduced in Section 2.5.5 followed this path by leveraging representation learning for modeling knowledge bases and proposed to use an auto-encoder model: their encoder extracts the relation from a sentence that the decoder uses to predict a missing entity. However, their encoder is still limited compared to its supervised counterpart (e.g. PCNN) and relies on handcrafted features extracted by natural language processing tools (Section 2.3.4). These features tend to contain errors and prevent the discovery of new patterns, which might hinder performances.

While the transition to deep learning approaches can bring more expressive models to the task, it also raises new problems. This chapter tackles a problem specific to unsupervised discriminative relation extraction

“ And once again I am I will not say alone, no, that’s not like me, but, how shall I say, I don’t know, restored to myself, no, I never left myself, free, yes, I don’t know what that means but it’s the word I mean to use, free to do what, to do nothing, to know, but what, the laws of the mind perhaps, of my mind, that for example water rises in proportion as it drowns you and that you would do better, at least no worse, to obliterate texts than to blacken margins, to fill in the holes of words till all is blank and flat and the whole ghastly business looks like what is, senseless, speechless, issueless misery.

— Samuel Beckett, *Molloy* (1955)

“ Careful! We don’t want to learn anything from this.

— Bill Watterson, *Calvin and Hobbes* (1992)

This chapter is an adaptation of an article published at ACL with some supplementary results:

Étienne Simon et al. (July 2019). “Unsupervised Information Extraction: Regularizing Discriminative Approaches with Relation Distribution Losses”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1378–1387. DOI: 10.18653/v1/P19-1133. URL: <https://www.aclweb.org/anthology/P19-1133>

4375 models. In particular, we focus on the VAE model of Section 2.5.5. These  
 4376 models tend to be hard to train because of the way  $\mathcal{H}_{\text{UNIFORM}}$  is enforced,  
 4377 expressly, how we ensure that all relations are conveyed the same amount  
 4378 of time.<sup>48</sup> To tackle this issue, we propose two new regularizing losses on  
 4379 the distribution of relations. With these, we hope to leverage the expres-  
 4380 sivity of discriminative approaches—in particular, of deep neural network  
 4381 classifiers—while staying in an unsupervised setting. Indeed, these models  
 4382 are hard to train without supervision, and the solutions proposed at the  
 4383 time were unstable. Discriminative approaches have less inductive bias,  
 4384 but this makes them more sensitive to noise.

4385 Indeed, our initial experiments showed that the VAE relation extraction  
 4386 model was unstable, especially when using a deep neural network relation  
 4387 classifier. It converges to either of the two following regimes, depending on  
 4388 hyperparameter settings: always predicting the same relation or predicting  
 4389 a uniform distribution. To overcome these limitations, we propose to use  
 4390 two new losses alongside an entity prediction loss based on a fill-in-the-  
 4391 blank task and show experimentally that this is key to learning deep neural  
 4392 network models. Our contributions are the following:

- 4393 • We propose two RelDist losses: a skewness loss, which encourages  
 4394 the classifier to predict a class with confidence for a single sentence,  
 4395 and a distribution distance loss, which encourages the classifier to  
 4396 scatter a set of sentences into different classes;
- 4398 • We perform extensive experiments on the usual NYT + FB dataset,  
 4399 as well as two new datasets;
- 4401 • We show that our RelDist losses allow us to train a deep PCNN  
 4402 classifier and improve the performances of feature-based models.

4404 In this chapter, we first describe our model in Section 3.1 before revis-  
 4405 iting the related works pertinent to the experimental setup in Section 3.2.  
 4406 We present our main experimental results in Section 3.3 before studying  
 4407 some possible improvements we considered in Section 3.4.

4408

4409

### 4410 3.1 Model description

4411

4412 Our model focuses on extracting the relation between two entities in tex-  
 4413 tual data and assumes that an entity chunker has identified named entities  
 4414 in the text. Furthermore, following Section 2.1, we limit ourselves to bi-  
 4415 nary relations and therefore consider sentences with two tagged entities,  
 4416 as shown in Figure 3.1. These sentences constitute the set  $\mathcal{S}$ . We further  
 4417 assume that entity linking was performed and that we have access to entity  
 4418 identifiers from the set  $\mathcal{E}$ . We therefore consider samples from a dataset  
 4419  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}^2$ . From these samples we learn a relation classifier that maps  
 4420 each sample  $x \in \mathcal{D}$  to a relation  $r \in \mathcal{R}$ . As such, our approach is sentential  
 4421 (Section 2.1).

4422 To provide a supervision signal to our relation classifier, we follow the  
 4423 VAE model of Section 2.5.5 (Marcheggiani and Titov 2016). However, the  
 4424 interpretation of their model as a VAE is part of the limitation we observed  
 4425 and is in conflict with the modifications we introduce. We, therefore, re-  
 4426 formulate their approach as a *fill-in-the-blank* task:

4427

4428

“The sol <sub>$e_1$</sub>  was the currency of ? <sub>$e_2$</sub>  between 1863 and 1985.”

<sup>48</sup> However, this problem can be gen-  
 eralized to how we enforce all relations  
 are conveyed reasonably often.



(PCNN, Section 2.3.6, Zeng et al. 2015). Similar to DIPRE’s split-in-three-affixes, the input sentence can be split into three parts separated by the two entities (see Figure 3.1). In a PCNN, the model outputs a representation for each part of the sentence. These are then combined to make a prediction. Figure 2.6 shows the network architecture that we now describe.

First, each word of  $s$  is mapped to a real-valued vector. In this work, we use standard word embeddings, initialized with GloVe<sup>49</sup> (Section 1.2.1, Pennington et al. 2014), and fine-tune them during training. Based on those embeddings, a convolutional layer detects patterns in subsequences of words. Then, a max-pooling along the text length combines all features into a fixed-size representation. Note that in our architecture, we obtained better results by using three distinct convolutions, one for each sentence part (i.e. the weights are not shared). We then apply a non-linear function (tanh) and sum the three vectors into a single representation for  $s$ . Finally, this representation is fed to a softmax layer to predict the distribution over the relations. This distribution can be plugged into Equation 3.1. Denoting PCNN our classifier, we have:

$$P(r | s) = \text{PCNN}(r; s, \phi),$$

where  $\phi$  are the parameters of the classifier. Note that we can use the PCNN to predict the relationship for any pair of entities appearing in any sentence since the input will be different for each selected pair (see Figure 2.6). Furthermore, since the PCNN ignore the entities surface forms, we can have  $P(r | s) = P(r | \text{blanked}(s))$  which is necessary to enforce  $\mathcal{H}_{\text{BLANKABLE}}$ .

### 3.1.2 Entity Predictor

The purpose of the entity predictor is to provide supervision for the relation classifier. As such, it needs to be differentiable. We follow Marchegiani and Titov (2016) to model  $P(e_i | r, e_{-i})$ , and use an energy-based formalism, where  $\psi(e_1, r, e_2)$  is the energy associated with  $(e_1, r, e_2)$ . The probability is obtained as follows:

$$P(e_1 | r, e_2) = \frac{\exp(\psi(e_1, r, e_2))}{\sum_{e' \in \mathcal{E}} \exp(\psi(e', r, e_2))}, \quad (3.2)$$

where  $\psi$  is expressed as the sum of two standard relational learning models: selectional preferences (Section 1.4.2.1) and RESCAL (Section 1.4.2.2):

$$\psi(e_1, r, e_2; \theta) = \underbrace{\mathbf{u}_{e_1}^\top \mathbf{a}_r + \mathbf{u}_{e_2}^\top \mathbf{b}_r}_{\text{Selectional Preferences}} + \underbrace{\mathbf{u}_{e_1}^\top \mathbf{C}_r \mathbf{u}_{e_2}}_{\text{RESCAL}}$$

where  $\mathbf{U} \in \mathbb{R}^{\mathcal{E} \times m}$  is an entity embedding matrix,  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{\mathcal{R} \times m}$  are two matrices encoding the preferences of each relation of certain entities,  $\mathbf{C} \in \mathbb{R}^{\mathcal{R} \times m \times m}$  is a three-way tensor encoding the entities interactions, and the hyperparameter  $m$  is the dimension of the embedded entities. The function  $\psi$  also depends on the energy functions parameters  $\theta = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}\}$  that we might omit for legibility. RESCAL (Nickel et al. 2011) uses a bilinear tensor product to gauge the compatibility of the two entities; whereas, in the Selectional Preferences model, only the predisposition of an entity to appear as the subject or object of a relation is captured.

**Negative Sampling** The number of entities being very large, the partition function of Equation 3.2 cannot be efficiently computed. To avoid

Zeng et al., “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks” EMNLP 2015

<sup>49</sup> We use the 6B.50d pre-trained word embeddings from <https://nlp.stanford.edu/projects/glove/>

the summation over the set of entities, we follow Section 1.2.1.3 and use negative sampling (Mikolov et al. 2013b); instead of training a softmax classifier, we train a discriminator which tries to recognize real triplets ( $D = 1$ ) from fake ones ( $D = 0$ ):

$$P(D = 1 \mid e_1, e_2, r) = \sigma(\psi(e_1, r, e_2)),$$

where  $\sigma(x) = 1 / (1 + \exp(-x))$  is the sigmoid function. This model is then trained by generating negative entities for each position and optimizing the negative log-likelihood:

$$\begin{aligned} \mathcal{L}_{\text{EP}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = & \mathbb{E}_{\substack{(s, e_1, e_2) \sim \mathcal{U}(\mathcal{D}) \\ r \sim \text{PCNN}(s; \boldsymbol{\phi})}} \left[ -\log \sigma(\psi(e_1, r, e_2; \boldsymbol{\theta}) + b_{e_1}) \right. \\ & -\log \sigma(\psi(e_1, r, e_2; \boldsymbol{\theta}) + b_{e_2}) \\ & - \sum_{j=1}^k \mathbb{E}_{e' \sim \mathcal{U}_{\mathcal{D}}(\mathcal{E})} [\log \sigma(-\psi(e_1, r, e'; \boldsymbol{\theta}) - b_{e'})] \\ & \left. - \sum_{j=1}^k \mathbb{E}_{e' \sim \mathcal{U}_{\mathcal{D}}(\mathcal{E})} [\log \sigma(-\psi(e', r, e_2; \boldsymbol{\theta}) - b_{e'})] \right] \end{aligned} \quad (3.3)$$

This loss is defined over the empirical data distribution  $\mathcal{U}(\mathcal{D})$ , i.e. the samples  $(s, e_1, e_2)$  follow a uniform distribution over sentences tagged with two entities; and the empirical entity distribution  $\mathcal{U}_{\mathcal{D}}(\mathcal{E})$ , that is the categorical distribution over  $\mathcal{E}$  where each entity is weighted by its frequency in  $\mathcal{D}$ . The distribution of the relation  $r$  for the sentence  $s$  is then given by the classifier  $\text{PCNN}(s; \boldsymbol{\phi})$ , which corresponds to the  $\sum_{r \in \mathcal{R}} P(r \mid s)$  in Equation 3.1. Following standard practice, during training, the expectation on negative entities is approximated by sampling  $k$  random entities following the empirical entity distribution  $\mathcal{E}$  for each position.

**Biases** Following Marcheggiani and Titov (2016), we add a bias for entities to  $\psi$ . These biases are parametrized by a single vector  $\mathbf{b} \in \mathbb{R}^{\mathcal{E}}$ . They encode how some entities are more likely to appear than others; as such, the  $+b_{e_i}$  appear in  $\mathcal{L}_{\text{EP}}$  where the  $P(e_i \mid r, e_{-i})$  would appear in the negative sampling estimation.

**Approximation** When  $|\mathcal{R}|$  is large, the expectation over  $r \sim \text{PCNN}(s; \boldsymbol{\phi})$  can be slow to evaluate. To avoid computing  $\psi$  for all possible relation  $r \in \mathcal{R}$ , we employ an optimization also used by Marcheggiani and Titov (2016). This optimization is built upon the following approximation:

$$\mathbb{E}_{r \sim \text{PCNN}(s; \boldsymbol{\phi})} [\log \sigma(\psi(e_1, r, e_2; \boldsymbol{\theta}))] \approx \log \sigma \left( \mathbb{E}_{r \sim \text{PCNN}(s; \boldsymbol{\phi})} [\psi(e_1, r, e_2; \boldsymbol{\theta})] \right). \quad (3.4)$$

Since the function  $\psi$  is linear in  $r$ , we can efficiently compute its expected value over  $r$  using the convex combinations of the relation embeddings. For example we can replace the selectional preference of a relation  $r$  for a head entity  $e_1$ :  $\mathbf{u}_{e_1}^T \mathbf{a}_r$  by the selectional preference of a distribution  $\text{PCNN}(s; \boldsymbol{\phi})$  for a head entity:  $\mathbf{u}_{e_1}^T (\text{PCNN}(s; \boldsymbol{\phi})^T \mathbf{A})$ .

### 3.1.3 RelDist losses

Training the classifier through Equation 3.3 alone is very unstable and dependent on precise hyperparameter tuning. More precisely, according to our early experiments, the training process usually collapses into one of two regimes:

- ( $\mathcal{P}1$ ) The classifier is very uncertain about which relation is expressed and outputs a uniform distribution over relations (Figure 3.2);
- ( $\mathcal{P}2$ ) All sentences are classified as conveying the same relation (Figure 3.3).

In both cases, the entity predictor can do a good job minimizing  $\mathcal{L}_{EP}$  by ignoring the output of the classifier, simply exploiting entities' co-occurrences. More precisely, many entities only appear in one relationship with a single other entity. In this case, the entity predictor can easily ignore the relationship  $r$  and predict the missing entity—and this pressure is even worse at the beginning of the optimization process as the classifier's output is not yet reliable.

This instability problem is particularly prevalent since the two components (classifier and entity predictor) are strongly interdependent: the classifier cannot be trained without a good entity predictor, which itself cannot take  $r$  into account without a good classifier resulting in a bootstrapping problem. To overcome these pitfalls, we developed two additional losses, which we now describe.

**Skewness.** Firstly, to encourage the classifier to be confident in its output, we minimize the entropy of the predicted relation distribution. This addresses  $\mathcal{P}1$  by forcing the classifier toward outputting one-hot vectors for a given sentence using the following loss:

$$\mathcal{L}_s(\phi) = \mathbb{E}_{(s,e) \sim \mathcal{U}(\mathcal{D})} [\mathbf{H}(\mathbf{R} \mid s, e; \phi)], \quad (3.5)$$

where  $\mathbf{R}$  is the random variable corresponding to the predicted relation. Following our first independence hypothesis, the entropy of equation 3.5 is equivalent to  $\mathbf{H}(\mathbf{R} \mid s)$ .

**Distribution Distance.** Secondly, to ensure that the classifier predicts several relations, we enforce  $\mathcal{H}_{UNIFORM}$  by minimizing the Kullback–Leibler divergence between the model prior distribution over relations  $P(\mathbf{R} \mid \phi)$  and the uniform distribution<sup>50</sup> over the set of relations  $\mathcal{U}(\mathcal{R})$ , that is:

$$\mathcal{L}_D(\phi) = \mathbf{D}_{KL}(P(\mathbf{R} \mid \phi) \parallel \mathcal{U}(\mathcal{R})). \quad (3.6)$$

Note that contrary to  $\mathcal{L}_s$ , to have a good approximation of  $P(\mathbf{R} \mid \phi)$ , the loss  $\mathcal{L}_D$  measures the unconditional distribution over  $\mathbf{R}$ , i.e. the distribution of predicted relations over all sentences. This addresses  $\mathcal{P}2$  by forcing the classifier toward predicting each class equally often over a set of sentences.

To satisfactorily and jointly train the entity predictor and the classifier, we use the two losses at the same time, resulting in the final loss:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{EP}(\theta, \phi) + \alpha \mathcal{L}_s(\phi) + \beta \mathcal{L}_D(\phi), \quad (3.7)$$

where  $\alpha$  and  $\beta$  are both positive hyperparameters.

Degenerate distributions:



⋮

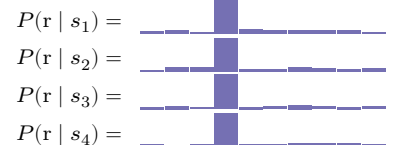
Desired distributions:



⋮

Figure 3.2: Illustration of  $\mathcal{P}1$ . The classifier assigns roughly the same probability to all relations. Instead, we would like the classifier to predict a single relation confidently.

Degenerate distributions:



⋮

average =



Desired distributions:



⋮

average =

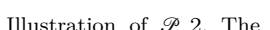


Figure 3.3: Illustration of  $\mathcal{P}2$ . The classifier consistently predicts the same relation. This is clearly visible when taking the average distribution (by marginalizing over the sentences  $s$ ). Instead, we would like the classifier to predict a diverse set of relations.

<sup>50</sup> Other distributions could be used, but in the absence of further information, this might be the best thing to do. See Section 3.5 for a discussion of alternatives.



All three losses are defined over the real data distribution, but in practice, they are approximated at the level of a mini-batch. First, both  $\mathcal{L}_{\text{EP}}$  and  $\mathcal{L}_s$  can be computed for each sample independently. To optimize  $\mathcal{L}_D$  however, we need to estimate  $P(R)$  at the mini-batch level and maximize the entropy of the mean predicted relation. Formally, let  $s_i$  for  $i = 1, \dots, B$  be the  $i$ -th sentence in a batch of size  $B$ , we approximate  $\mathcal{L}_D$  as:

$$\sum_{r \in \mathcal{R}} \left( \sum_{i=1}^B \frac{\text{PCNN}(r; s_i)}{B} \right) \log \left( \sum_{i=1}^B \frac{\text{PCNN}(r; s_i)}{B} \right).$$

**Learning** We optimize the empirical estimation of Equation 3.7, learning the PCNN parameters and word embeddings  $\phi$  as well as the entity predictor parameters and entity embeddings  $\theta$  jointly.

## 3.2 Related Work

The NLP and knowledge base related work is presented in Chapter 1, and the relation extraction related work is presented in Chapter 2. The main approaches we built upon are:

- Distant supervision (Section 2.2.2, Mintz et al. 2009): the method we use to obtain a supervised dataset for evaluation,<sup>51</sup>
- PCNN (Section 2.3.6, Zeng et al. 2015): our relation classifier, which was the state-of-the-art supervised relation extraction method at the time;
- Rel-LDA (Section 2.5.4, Yao et al. 2011): the state-of-the-art generative model we compare to;
- VAE for relation extraction (Section 2.5.5, Marcheggiani and Titov 2016): the overall inspiration for the architecture of our model, with which we share the entity predictor;
- SelfORE (Section 2.5.7, X. Hu et al. 2020): an extension of our work, which, alongside their own approach, proposed an improvement of our relation classifier by replacing the PCNN by a BERTcoder.

In this section, we give further details about the relationship between our losses and the ones derived by Marcheggiani and Titov (2016). As a reminder, their model is a VAE defined from an encoder  $Q(r | e, s; \phi)$  and a decoder  $P(e | r, s; \theta)$  as:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = \mathbb{E}_{Q(r|e,s;\phi)} [-\log P(e | r, s; \theta)] + \beta D_{\text{KL}}(Q(r | e, s; \phi) \| P(r | \theta)) \quad (3.8)$$

This is simply a rewriting of the ELBO of Equation 2.8 substituting relation extraction variables to the generic ones. There is however two differences compared to a standard VAE. First, the variable  $s$  is not reconstructed, it simply conditions the whole process. Second, the regularization term is weighted by a hyperparameter  $\beta$ . This makes the model of Marcheggiani and Titov (2016) a conditional  $\beta$ -VAE (Higgins et al. 2017; Sohn et al. 2015). The first summand of Equation 3.8 is called the reconstruction loss since it reconstructs the input variable  $e$  from the latent variable  $r$  and the conditional variable  $s$ . Since we followed the same structure for our

<sup>51</sup> As explained in Section 2.5.1.1, this is sadly standard in the evaluation of clustering approaches.

The prior of a conditional VAE  $P(r | \theta)$  is usually conditioned on  $s$  too. However, this additional variable is not used by Marcheggiani and Titov (2016).

Higgins et al., “ $\beta$ -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework” ICLR 2017  
Sohn et al., “Learning Structured Output Representation using Deep Conditional Generative Models” NeurIPS 2015

4699 model, this reconstruction loss is actually  $\mathcal{L}_{\text{EP}}$ , the difference being in the  
 4700 relation classifier. We can then rewrite the loss of Marcheggiani and Titov  
 4701 (2016) as:

$$\begin{aligned} 4702 \quad \mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathcal{L}_{\text{EP}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \beta \mathcal{L}_{\text{VAE REG}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ 4703 \quad \mathcal{L}_{\text{VAE REG}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= D_{\text{KL}}(Q(\mathbf{r} \mid \mathbf{e}; \boldsymbol{\phi}) \parallel P(\mathbf{r} \mid \boldsymbol{\theta})) \end{aligned}$$

As explained Section 2.5.5,  $Q$  is the VAE’s encoder.

4706 In their work, they select the prior as a uniform distribution over all rela-  
 4707 tions  $P(\mathbf{r} \mid \boldsymbol{\theta}) = \mathcal{U}(\mathcal{R})$  and approximate  $\mathcal{L}_{\text{VAE REG}}$  as follow:

$$4708 \quad \mathcal{L}_{\text{VAE REG}}(\boldsymbol{\phi}) = \mathbb{E}_{(s, \mathbf{e}) \sim \mathcal{U}(\mathcal{D})} [-\text{H}(\mathbf{R} \mid s, \mathbf{e}; \boldsymbol{\phi})]$$

4711 Its purpose is to prevent the classifier from always predicting the same  
 4712 relation, i.e. it has the same purpose as our distance loss  $\mathcal{L}_{\text{D}}$ . However, its  
 4713 expression is equivalent to  $-\mathcal{L}_{\text{S}}$ , and indeed, minimizing the opposite of our  
 4714 skewness loss increases the entropy of the classifier output, addressing  $\mathcal{P}2$   
 4715 (classifier always outputting the same relation). Yet, using  $\mathcal{L}_{\text{VAE REG}} = -\mathcal{L}_{\text{S}}$   
 4716 alone, draws the classifier into the other pitfall  $\mathcal{P}1$  (not predicting any  
 4717 relation confidently). In a traditional VAE,  $\mathcal{P}1$  is addressed by the recon-  
 4718 struction loss  $\mathcal{L}_{\text{EP}}$ . However, at the beginning of training, the supervision  
 4719 signal is so weak that we cannot rely on  $\mathcal{L}_{\text{EP}}$  for our task. The  $\beta$  weighting  
 4720 can be decreased to avoid  $\mathcal{P}1$ , but this would also lessen the solution to  
 4721  $\mathcal{P}2$ . This causes a drop in performance, as we show experimentally.

## 4724 3.3 Experiments

4726 To compare with previous works, we repeat the experimental setup of  
 4727 Marcheggiani and Titov (2016) with the B<sup>3</sup> evaluation metric (Bagga  
 4728 and Baldwin 1998). We complemented this setup with two additional  
 4729 datasets extracted from T-REX (Elsahar et al. 2018) and two more metrics  
 4730 commonly seen in clustering task evaluation: V-measure (Rosenberg and  
 4731 Hirschberg 2007) and ARI (Hubert and Arabie 1985). This allows us to  
 4732 capture the characteristics of each approach in more detail.

4733 In this section, we begin by describing the processing of the datasets  
 4734 in Section 3.3.1. We then describe the experimental details of the models  
 4735 we evaluated in Section 3.3.2. Finally, we give quantitative results in Sec-  
 4736 tion 3.3.3 and qualitative results in Section 3.3.4. The description of the  
 4737 metrics can be found in Section 2.5.1.1. Appendix C gives further details  
 4738 on the source datasets, their specificities, their sizes and some example of  
 4739 their content when appropriate.

### 4742 3.3.1 Datasets

4744 As explained in Section 2.5.1, to evaluate the models, we use labeled  
 4745 datasets, the labels being used for validation and testing. The first dataset  
 4746 we consider is the one of Marcheggiani and Titov (2016), which is similar  
 4747 to the one used in Yao et al. (2011). This dataset was built through distant  
 4748 supervision (Section 2.2.2) by aligning sentences from the New York Times  
 4749 corpus (NYT, Section C.5, Sandhaus 2008) with Freebase (FB, Section C.3,  
 4750 Bollacker et al. 2008) facts. Several sentences were filtered out based on  
 4751 features like the length of the dependency path between the two entities,  
 4752 resulting in 2 million sentences with only 41 000 (2%) of them labeled with

4753 one of 262 possible relations. 20% of the labeled sentences were set aside  
 4754 for validation; the remaining 80% are used to compute the final results.

4755 We also extracted two datasets from T-REX (Section C.7, Elsahar et  
 4756 al. 2018), which was built as an alignment of Wikipedia with Wikidata  
 4757 (Section C.8, Vrandečić and Kröttsch 2014). We only consider  $(s, e_1, e_2)$   
 4758 triplets where both entities appear in the same sentence.<sup>52</sup> If a single sen-  
 4759 tence contains multiple triplets, it appears multiple times in the dataset,  
 4760 each time with a different pair of tagged entities. We built the first dataset  
 4761 DS by extracting all triplets of T-REX where the two entities are linked by  
 4762 a relation in Wikidata. This is the usual distant supervision method. It re-  
 4763 sults in 1 189 relations and nearly 12 million sentences, all of them labeled  
 4764 with a relation.

4765 In Wikidata, each relation is annotated with a list of associated surface  
 4766 forms; for example, “*shares border with*” can be conveyed by “borders,”  
 4767 “adjacent to,” “next to,” etc. The second dataset we built, SPO, only con-  
 4768 tains the sentences where a surface form of the relation also appears in  
 4769 the sentence, resulting in 763 000 samples (6% of the unfiltered dataset)  
 4770 and 615 relations. This dataset still contains some misalignment, but it  
 4771 should nevertheless be easier for models to extract the correct semantic  
 4772 relation since the set of surface forms is much more restricted and much  
 4773 more regular.

4774

4775

### 4776 3.3.2 Baselines and Models

4777 We compare our model with three state-of-the-art approaches, two gener-  
 4778 ative rel-LDA models of Yao et al. (2011), the VAE model of Marcheggiani  
 4779 and Titov (2016) and the deep clustering of BERT representations by X.  
 4780 Hu et al. (2020).

4781 The two rel-LDA models only differ by the number of features consid-  
 4782 ered. We use the eight features listed in Marcheggiani and Titov (2016):  
 4783

4784

4785

4786

4787

4788

4789

4790

4791

4792

4793

4794

4795

4796

4797

4798 Rel-LDA uses the first three features, while rel-LDA1 is trained by iteratively  
 4799 adding more features until all eight are used.

4800 To assess our two main contributions individually, we evaluate the  
 4801 PCNN classifier and our additional losses separately. More precisely, we first  
 4802 study the effect of the RelDist losses by looking at the differences between  
 4803 models optimizing  $\mathcal{L}_{EP} + \mathcal{L}_{VAE\ REG}$  and the ones optimizing  $\mathcal{L}_{EP} + \mathcal{L}_S + \mathcal{L}_D$   
 4804 with  $\mathcal{L}_{EP}$  being either computed using the relation classifier of Marcheg-  
 4805 giani and Titov (2016) or our PCNN. Second, we study the effect of the  
 4806 relation classifier by comparing the feature-based classifier and the PCNN

<sup>52</sup> T-REX provides annotations for whole articles; it should therefore be possible to process broader contexts by defining  $\mathcal{S}$  as a set of articles. However, in this work, we stay in the traditional sentence-level relation extraction setup.

4807 trained with the same losses. We also give results for our RelDist losses  
 4808 together with a BERTcoder classifier. This latter combination is evaluated  
 4809 by X. Hu et al. (2020) following our experimental setup. We thus focus  
 4810 mainly on four models:

- 4811 • Linear +  $\mathcal{L}_{\text{VAE REG}}$ , which corresponds to the model of Marcheggiani  
 4812 and Titov (2016);
- 4813
- 4814 • Linear +  $\mathcal{L}_s + \mathcal{L}_D$ , which uses the feature-based linear encoder of  
 4815 Marcheggiani and Titov (2016) together with our RelDist losses;
- 4816
- 4817 • PCNN +  $\mathcal{L}_{\text{VAE REG}}$ , which uses our PCNN encoder together with the  
 4818 regularization of Marcheggiani and Titov (2016);
- 4819
- 4820 • PCNN +  $\mathcal{L}_s + \mathcal{L}_D$ , which is our complete model.
- 4821

4822 All models are trained with ten relation classes, which, while lower than  
 4823 the number of actual relations, allows us to compare the models faithfully  
 4824 since the distribution of gold relations is very unbalanced. For feature-  
 4825 based models, the size of the features domain range from 1 to 10 million  
 4826 values depending on the dataset. We train our models with Adam using  $L_2$   
 4827 regularization on all parameters. To have a good estimation of  $P(R)$  in the  
 4828 computation of  $\mathcal{L}_D$ , we use a batch size of 100. Our word embeddings are  
 4829 of size 50, entities embeddings of size  $m = 10$ . We sample  $k = 5$  negative  
 4830 samples to estimate  $\mathcal{L}_{\text{EP}}$ . Lastly, we set  $\alpha = 0.01$  and  $\beta = 0.02$ . All three  
 4831 datasets come with a validation set, and following Marcheggiani and Titov  
 4832 (2016), we used it for cross-validation to optimize the  $B^3 F_1$ .  
 4833

### 4834 3.3.3 Results

4835 The results reported in Table 3.1 are the average test scores of three runs  
 4836 on the NYT + FB and T-REX SPO datasets, using different random initial-  
 4837 ization of the parameters—in practice, the variance was low enough so  
 4838 that reported results can be analyzed. We observe that regardless of the  
 4839 model and metrics, the highest measures are obtained on T-REX SPO, then  
 4840 NYT + FB and finally T-REX DS. This was to be expected since T-REX SPO  
 4841 was built to be easy, while hard-to-process sentences were filtered out of  
 4842 NYT + FB (Marcheggiani and Titov 2016; Yao et al. 2011). We also observe  
 4843 that the main metrics agree in general ( $B^3$ , V-measure and ARI) in most  
 4844 cases. Performing a PCA on the measures, we observed that V-measure  
 4845 forms a nearly-orthogonal axis to  $B^3$ , and to a lesser extent ARI. Hence we  
 4846 can focus on  $B^3$  and V-measure in our analysis.  
 4847

4848 We first measure the benefit of our RelDist losses: on all datasets and  
 4849 metrics, the two models using  $\mathcal{L}_s + \mathcal{L}_D$  are systematically better than the  
 4850 ones using  $\mathcal{L}_{\text{VAE REG}}$ :  
 4851

- 4852 • The PCNN models consistently gain between 7 and 11 points in  $B^3$   
 4853  $F_1$  from these additional losses;
- 4854
- 4855 • The feature-based linear classifier benefits from the RelDist losses to  
 4856 a lesser extent, except on the T-REX DS dataset on which the Linear +  
 4857  $\mathcal{L}_{\text{VAE REG}}$  model without the RelDist losses completely collapses—we  
 4858 hypothesize that this dataset is too hard for the model given the  
 4859 number of parameters to estimate.  
 4860

Dataset	Model		B <sup>3</sup>			V-measure			ARI
	Classifier	Reg.	$F_1$	Prec.	Rec.	$F_1$	Hom.	Comp.	
NYT + FB	rel-LDA		29.1	24.8	35.2	30.0	26.1	35.1	13.3
	rel-LDA1		36.9	30.4	47.0	37.4	31.9	45.1	24.2
	Linear	$\mathcal{L}_{\text{VAE REG}}$	35.2	23.8	67.1	27.0	18.6	49.6	18.7
	PCNN	$\mathcal{L}_{\text{VAE REG}}$	27.6	24.3	31.9	24.7	21.2	29.6	15.7
	Linear	$\mathcal{L}_S + \mathcal{L}_D$	37.5	31.1	47.4	<b>38.7</b>	32.6	47.8	27.6
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	<b>39.4</b>	32.2	50.7	38.3	32.2	47.2	<b>33.8</b>
	BERTcoder <sup>†</sup>	$\mathcal{L}_S + \mathcal{L}_D$	41.5	34.6	51.8	39.9	33.9	48.5	35.1
	BERTcoder <sup>†</sup>	SelfORE <sup>†</sup>	<i>49.1</i>	47.3	51.1	<i>46.6</i>	45.7	47.6	<i>40.3</i>
T-REX SPO	rel-LDA		11.9	10.2	14.1	5.9	4.9	7.4	3.9
	rel-LDA1		18.5	14.3	26.1	19.4	16.1	24.5	8.6
	Linear	$\mathcal{L}_{\text{VAE REG}}$	24.8	20.6	31.3	23.6	19.1	30.6	12.6
	PCNN	$\mathcal{L}_{\text{VAE REG}}$	25.3	19.2	37.0	23.1	18.1	31.9	10.8
	Linear	$\mathcal{L}_S + \mathcal{L}_D$	29.5	22.7	42.0	34.8	28.4	45.1	20.3
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	<b>36.3</b>	28.4	50.3	<b>41.4</b>	33.7	53.6	<b>21.3</b>
	BERTcoder <sup>†</sup>	$\mathcal{L}_S + \mathcal{L}_D$	38.1	30.7	50.3	39.1	37.6	40.8	23.5
	BERTcoder <sup>†</sup>	SelfORE <sup>†</sup>	<i>41.0</i>	39.4	42.8	<i>41.4</i>	40.3	42.5	<i>33.7</i>
T-REX DS	rel-LDA		9.7	6.8	17.0	8.3	6.6	11.4	2.2
	rel-LDA1		12.7	8.3	26.6	17.0	13.3	23.5	3.4
	Linear	$\mathcal{L}_{\text{VAE REG}}$	9.0	6.4	15.5	5.7	4.5	7.9	1.9
	PCNN	$\mathcal{L}_{\text{VAE REG}}$	12.2	8.6	21.1	12.9	10.1	18.0	2.9
	Linear	$\mathcal{L}_S + \mathcal{L}_D$	19.5	13.3	36.7	<b>30.6</b>	24.1	42.1	<b>11.5</b>
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	<b>19.7</b>	14.0	33.4	26.6	20.8	36.8	9.4
	BERTcoder <sup>†</sup>	$\mathcal{L}_S + \mathcal{L}_D$	22.4	17.6	30.8	31.2	26.3	38.3	12.3
	BERTcoder <sup>†</sup>	SelfORE <sup>†</sup>	<i>32.9</i>	29.7	36.8	<i>32.4</i>	30.1	35.1	<i>20.1</i>

Table 3.1: Results (percentage) on our three datasets. The results for rel-LDA, rel-LDA1, Linear and PCNN are our own, while results for BERTcoder and SelfORE, marked with <sup>†</sup>, are from X. Hu et al. (2020). The best results at the time of publication of our article are in **bold**, while the best results at the time of writing are in *italic*.

We now restrict to discriminative models based on  $\mathcal{L}_S + \mathcal{L}_D$ . We note that both relation classifier (Linear and PCNN) exhibit better performances than generative ones (rel-LDA, rel-LDA1) with a difference ranging from 2.5/0.6 (NYT + FB, for Linear/PCNN) to 11/17.8 (on T-REX SPO). However, the advantage of PCNNs over feature-based classifiers is not completely clear. While the PCNN version has a systematically better B<sup>3</sup>  $F_1$  on all datasets (differences of 1.9/6.8/0.2 respectively for NYT+FB/T-REX SPO/T-REX DS), the V-measure decreases by 0.4/4.0 on respectively NYT + FB/T-REX DS, and ARI by 2.1 on T-REX DS. As B<sup>3</sup>  $F_1$  was used for validation, this shows that the PCNN models overfit this metric by polluting relatively clean clusters with unrelated sentences or degrades well clustered gold relations by splitting them into two clusters.

The BERTcoder classifier improves all metrics consistently, with the sole exception of the V-measure on the T-REX SPO dataset. This can be explained both by the larger expressive power of BERT and by its pretraining as a language model. The SelfORE model, which is built on top of a BERTcoder further improves the results on all datasets. Since these results are from a subsequent work (X. Hu et al. 2020), we won't delve too much into details. As mentioned in Section 2.5.7, SelfORE is an iterative algorithm; the  $\mathcal{R}_{\text{UNIFORM}}$  assumption is enforced on the whole dataset at once, thus solving  $\mathcal{P}2$ . While to solve  $\mathcal{P}1$ , SelfORE uses a concentration objective (through the square in the target distribution  $\mathbf{P}$  in Equation 2.11). While the BERTcoder can replace our PCNN classifier and can be evaluated

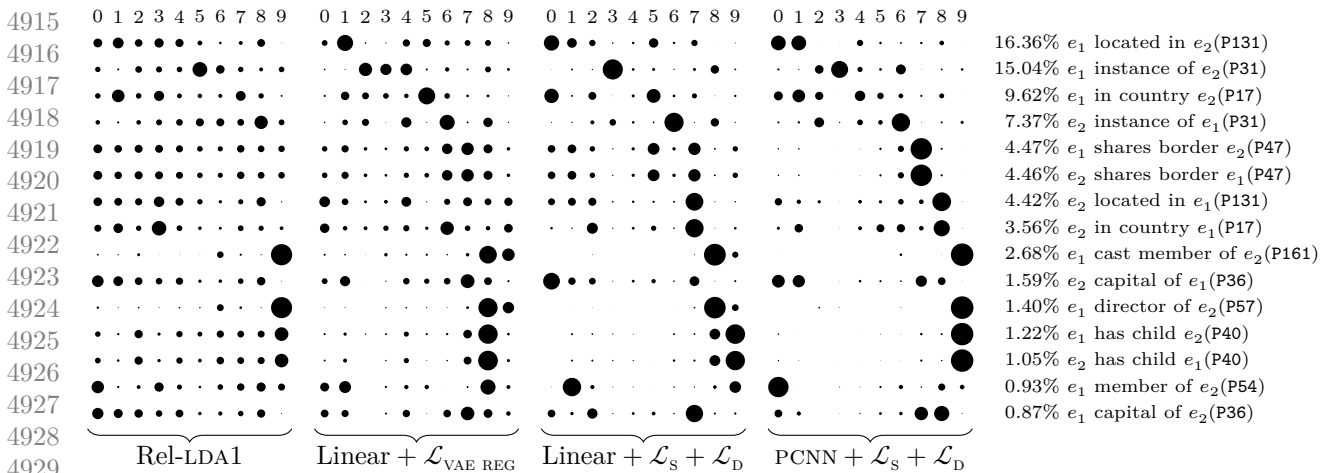


Figure 3.4: Normalized confusion matrices for the T-REX SPO dataset. For each model, each of the 10 columns corresponds to a predicted relation cluster, which were sorted to ease comparison. The rows identify Wikidata relations sorted by their frequency in the T-REX SPO corpus (reported as percentage in front of each relation name). The area of each circle is proportional to the number of sentences in the cell. For clarity, the matrix was normalized so that each row sum to 1, thus it is more akin to a  $B^3$  per-item recall than a true confusion matrix.

with our regularization losses, the SelfORE algorithm is a replacement for the  $\mathcal{L}_{\text{EP}} + \mathcal{L}_S + \mathcal{L}_D$  and can't be use jointly with  $\mathcal{L}_S + \mathcal{L}_D$ . In theory, the SelfORE algorithm could be used with a linear or PCNN encoder. However, SelfORE strongly relies on a good initial representation; such a model would need to be pre-trained as a language model beforehand.

### 3.3.4 Qualitative Analysis

Since, for our model of interest, all the metrics agree on the T-REX SPO dataset, we plot the confusion matrix of our models in Figure 3.4. Each row is labeled with the gold Wikidata relation extracted through distant supervision. For example, the top left cell of each matrix correspond to the value  $P(c(X) = 0 \mid g(X) = "e_1 \text{ located in } e_2")$  using the notation of Section 2.5.1. Since relations are generally not symmetric, each Wikidata relation appears twice in the table, once for each disposition of the entities in the sentence. This is particularly problematic with symmetric relations such as "shares border," which are two different gold relations that actually convey the same semantic relation.

To interpret Figure 3.4, we have to see whether a predicted cluster (column) contains different gold relations—paying attention to the fact that the most important gold relations are listed in the top rows (the top 5 relations account for 50% of sentences). The first thing to notice is that the confusion matrix of both models using our RelDist losses ( $\mathcal{L}_S + \mathcal{L}_D$ ) are sparser (for each column), which means that our models better separate relations from each other. We observe that Linear +  $\mathcal{L}_{\text{VAE REG}}$  (the model of the model of Marcheggiani and Titov 2016) is affected by the pitfall  $\mathcal{P}1$  (uniform distribution) for many gold clusters. The  $\mathcal{L}_{\text{VAE REG}}$  loss forces the classifier to be uncertain about which relation is expressed, translating into a dense confusion matrix and resulting in poor performances. The rel-LDA1 model is even worse and fails to identify clear clusters, showing the limitations of a purely generative approach that might focus on features not linked with any relation.

Focusing on our proposed model, PCNN +  $\mathcal{L}_S + \mathcal{L}_D$  (rightmost figure), we looked at two different mistakes. The first is a gold cluster divided in two (low recall). When looking at clusters 0 and 1, we did not find any recognizable pattern. Moreover, the corresponding entity predictor parameters are very similar. This seems to be a limitation of the distance loss: splitting a large cluster in two may improve  $\mathcal{L}_D$  but worsen all the evaluation metrics.

4969 The model is then penalized by the fact that it lost one slot to transmit  
 4970 information between the classifier and the entity predictor. The second  
 4971 type of mistake is when a predicted cluster corresponds to two gold ones  
 4972 (low precision). Here, most of the mistakes seem understandable: “shares  
 4973 border” is symmetric (cluster 7), “located in” and “in country” (cluster 8)  
 4974 or “cast member” and “director of” (cluster 9) are clearly related. Note  
 4975 that other variants are also affected similarly, showing that the problem  
 4976 of granularity is complex.

4977  
 4978

### 4979 3.4 Alternative Models

4980

4981 In this section, we present some variations we considered during the devel-  
 4982 opment of our model. However, we did not manage to obtain satisfactory  
 4983 results with these variants. When possible, we provide an analysis of why  
 4984 we think these variants did not work; keeping in mind that negative re-  
 4985 sults are difficult to certify, poor results might be improved with a better  
 4986 hyperparameter search.

4987

4988 **LSTM Relation Classifier** Instead of a PCNN, we tried using a deep  
 4989 LSTM (Section 1.3.2.1) for our relation classifier. We never managed to  
 4990 obtain any results with them; the training always collapsed into one of  
 4991  $\mathcal{P}1$  or  $\mathcal{P}2$ . An LSTM is quite a lot harder to train than a CNN. The repre-  
 4992 sentation provided by an LSTM is the result of several non-linear operator  
 4993 compositions, through which it is hard to backpropagate information. On  
 4994 the other hand, with good initialization, the representation extracted by  
 4995 a CNN can be close to its input embeddings (which are pre-trained). Since  
 4996 the training of the entity predictor heavily depends on the relation classi-  
 4997 fier, it is not surprising that the training fails with an LSTM. The failure of  
 4998 the LSTM to provide a good representation at the beginning of the train-  
 4999 ing procedure pushes the entity predictor to ignore the relation variable  $r$ ,  
 5000 which therefore does not receive any gradient and thus does not provide  
 5001 any supervision back to the LSTM. Retrospectively, pre-training the sen-  
 5002 tence representation extractor with a language modeling loss could have  
 5003 overcome this problem. The initial representation would have been good  
 5004 enough for the entity predictor to provide some gradient back to the rela-  
 5005 tion classifier. This is confirmed by the work of X. Hu et al. (2020), who  
 5006 trained a BERT relation classifier with our losses. In the end, what made a  
 5007 PCNN work is its shallowness and the pre-trained GloVe word embeddings.

5008  
 5009 **Gumbel-Softmax** Another approach to tackling  $\mathcal{P}1$  (uniform out-  
 5010 put) would be to use a discrete distribution for the relation  $r$ ; instead  
 5011 of marginalizing over all possible relations in Equation 3.3, we would only  
 5012 take the most likely relation. However, taking the maximum would not be  
 5013 differentiable. The Gumbel-softmax technique provides a solution to this  
 5014 problem. Let’s call  $y_r \in \mathbb{R}$  for  $r \in \mathcal{R}$  the unnormalized score assigned to  
 5015 each relation by the PCNN. It can be shown (Gumbel 1954) that sampling  
 5016 from  $\text{softmax}(\mathbf{y})$  is equivalent to taking  $\text{argmax}_{r \in \mathcal{R}} y_r + G_r$  where  $G_r$  are  
 5017 randomly sampled from the Gumbel distribution. Knowing this, Jang et  
 5018 al. (2016) propose to use the following Gumbel-Softmax distribution:

5020

5021

5022

$$\pi_r = \frac{(\exp(y_r) + G_r) / \tau}{\sum_{r' \in \mathcal{R}} (\exp(y_{r'}) + G_{r'}) / \tau}$$

	$\mathcal{P}1$ solution	$B^3$			V-measure			ARI
		$F_1$	Prec.	Rec.	$F_1$	Hom.	Comp.	
5023								
5024								
5025								
5026	$\mathcal{L}_s$ regularization	39.4	32.2	50.7	38.3	32.2	47.2	33.8
5027	Gumbel–Softmax	35.0	29.9	42.2	33.2	28.3	40.2	25.1

Table 3.2: Quantitative results of the Gumbel–Softmax model on the NYT + FB dataset. The  $\mathcal{L}_s$  solution is used together with  $\mathcal{L}_D$  and a softmax activation, while the Gumbel–Softmax activation is used with  $\mathcal{L}_D$  only. Therefore, the first row reports the same results present in Table 3.1.

5030 This distribution has the advantage of being differentiable, barring the  
5031 Gumbel variables  $G_r$ . Furthermore, when the temperature  $\tau > 0$  is close  
5032 to 1, this distribution looks like a standard softmax output. On the other  
5033 hand, when the temperature is close to 0, this distribution is closer to  
5034 a one-hot vector with low entropy. Decreasing the temperature gradually  
5035 throughout the training process, this should help us solve  $\mathcal{P}1$ .

5036 Following a grid search, we initially set  $\tau = 1$  with an annealing rate  
5037 of 0.9 per epoch. Table 3.2 compares the best Gumbel–Softmax results of  
5038  $\mathcal{L}_{EP} + \mathcal{L}_D$  with the standard softmax result of  $\mathcal{L}_{EP} + \mathcal{L}_s + \mathcal{L}_D$  discussed above.  
5039 We do not use  $\mathcal{L}_s$  with Gumbel–Softmax since both mechanisms seek to  
5040 address  $\mathcal{P}1$ . While the Gumbel–Softmax prevents the model from falling  
5041 entirely into  $\mathcal{P}1$ , it still underperforms compared to the  $\mathcal{L}_s$  regularization  
5042 of our standard model.

5043  
5044 **Aligning Sentences and Entity Pairs** Another model we attempted  
5045 to train purposes to align sentences and entities. It recombines our PCNN  
5046 relation classifier with the energy function  $\psi$  into a new layout following a  
5047 relaxation of the  $\mathcal{H}_{PULLBACK}$  assumption.<sup>53</sup> In this model, we obtain a dis-  
5048 tribution over the relations  $P(r_s | \text{blanked}(s))$  using a PCNN as described  
5049 Section 3.1.1, but we also extract a distribution  $P(r_e | e)$  using the energy  
5050 function  $\psi$  normalized over the relations  $P(r_e | e_1, e_2) \propto \exp(\psi(e_1, r_e, e_2))$ .  
5051 This model clearly assumes  $\mathcal{H}_{PULLBACK}$  since it extracts a relation from the  
5052 entities and from the sentence separately. However, in contrast to other  
5053 models assuming  $\mathcal{H}_{PULLBACK}$  (such as DIPRE, Section 2.3.2), we combine the  
5054 separate relations into a single one to express the fact that a relation is  
5055 both conveyed by the sentence and the entities:

$$5056 \quad P(r = r | s, e; \theta, \phi) = P(r_s = r | s; \phi)P(r_e = r | e; \theta) \quad (3.9)$$

5058 For the final prediction  $r$ , the assumption  $\mathcal{H}_{PULLBACK}$  is not made, since it  
5059 depends both on the sentence and entities. However, Equation 3.9 clearly  
5060 assumes that  $r_s$  and  $r_e$  are independent and  $r$  does not capture any inter-  
5061 action between  $s$  and  $e$ . To train this model, we force the two distributions  
5062 to align by maximizing:  
5063

$$5064 \quad \mathcal{L}_{ALIGN}(\theta, \phi) = -\log \sum_{r \in \mathcal{R}} P(r | s, e; \theta, \phi) + \mathcal{L}_D(\theta) + \mathcal{L}_D(\phi). \quad (3.10)$$

5066 Here  $\mathcal{L}_s$  is not needed since, in order to maximize the pointwise product  
5067 of two probability mass functions, each distribution must be deterministic  
5068 on a matching relation, which solves  $\mathcal{P}1$ .

5069 Table 3.3 gives the results on the NYT + FB datasets and compares  
5070 them to the fill-in-the-blank model of Section 3.1. The main problem we  
5071 have with this model is its lack of stability. The average, maximum and  
5072 minimum given in Table 3.3 are computed over eight runs. Similar results  
5073 were observed with slightly different setups such as enforcing  $\mathcal{L}_D$  on the  
5074 product ( $r$ ) instead of each distribution separately ( $r_s$  and  $r_e$ ). As we  
5075 can see, the alignment model sometimes reaches excellent performances  
5076

<sup>53</sup> This hypothesis introduced Section 2.2.1 assumes that the relation can be found from the entities alone, and from the relations alone.

For numerical stability, the first term of Equation 3.10 needs to be computed as:

$$\begin{aligned} -\log \sum_{r \in \mathcal{R}} P(r | s, e; \theta, \phi) = \\ -\log \sum_{r \in \mathcal{R}} \exp(y_r^{(s)} + y_e^{(s)}) \\ + \log \sum_{r \in \mathcal{R}} \exp(y_r^{(s)}) \\ + \log \sum_{r \in \mathcal{R}} \exp(y_r^{(e)}) \end{aligned}$$

where  $y^{(s)}$  and  $y^{(e)}$  are the logits used for predicting  $r_s$  and  $r_e$  respectively.

We also attempted (without success) to align the two distribution by minimizing  $D_{JSD}(r_s || r_e)$ . Where  $D_{JSD}$  is the Jensen–Shannon divergence defined as:

$$D_{JSD}(r_s || r_e) = \frac{1}{2} (D_{KL}(r_s || m) + D_{KL}(r_e || m))$$

with  $P(m) = \frac{1}{2}(P(r_s) + P(r_e))$ .



Model	B <sup>3</sup>			V-measure			ARI
	$F_1$	Prec.	Rec.	$F_1$	Hom.	Comp.	
$\mathcal{L}_{EP} + \mathcal{L}_s + \mathcal{L}_D$	39.4	32.2	50.7	38.3	32.2	47.2	33.8
$\mathcal{L}_{ALIGN}$ average	37.6	30.3	49.7	39.4	33.1	48.8	20.3
$\mathcal{L}_{ALIGN}$ maximum	41.2	33.6	53.4	43.5	36.9	53.1	29.5
$\mathcal{L}_{ALIGN}$ minimum	34.5	26.5	49.3	35.9	29.6	45.7	15.3

Table 3.3: Quantitative results of the alignment model on the NYT + FB dataset. The first row reports the same results present in Table 3.1. Eight alignment models were trained, the average scores are given in the second row, while the third and fourth rows report the best and worst model among the eight.

relative to the fill-in-the-blank model. However, this happens rarely, and on average, it performs more poorly according to the B<sup>3</sup> and ARI metrics. Its good V-measures scores are nevertheless encouraging.

## 3.5 Conclusion

In this chapter, we show that discriminative relation extraction models can be trained efficiently on unlabeled datasets. Unsupervised relation extraction models tend to produce impure clusters by enforcing a uniformity constraint at the level of a single sample. We proposed two losses (named RelDist) to effectively train expressive relation extraction models by enforcing the distribution over relations to be uniform—note that other target distributions could be used. In particular, we were able to successfully train a deep neural network classifier that only performed well in a supervised setting so far. We demonstrated the effectiveness of our RelDist losses on three datasets and showcased its effect on cluster purity.

While forcing a uniform distribution with the distance loss  $\mathcal{L}_D$  might be meaningful with a low number of predicted clusters, it might not generalize to larger numbers of relations. Preliminary experiments seem to indicate that this can be addressed by replacing the uniform distribution in Equation 3.6 with the empirical distribution of the relations in the validation set or any other appropriate law if no validation set is available.<sup>54</sup> This would allow us to avoid the  $\mathcal{H}_{UNIFORM}$  assumption.

All models presented in this chapter make extensive independence assumptions. As inferred in Section 3.4 and shown in subsequent work (X. Hu et al. 2020; Soares et al. 2019), this could be solved with sentence representations pre-trained with a language modeling task. Furthermore, the fill-in-the-blank model is inherently sentence-level. In the next chapter, we study how to build an unsupervised aggregate relation extraction model using a pre-trained BERTcoder.

<sup>54</sup> In practice, Zipf’s law (described in the margin of Section 2.5.2) seems to fit the observed empirical distribution quite well.

5131  
5132  
5133  
5134  
5135  
5136  
5137  
5138  
5139  
5140  
5141  
5142  
5143  
5144  
5145  
5146  
5147  
5148  
5149  
5150  
5151  
5152  
5153  
5154  
5155  
5156  
5157  
5158  
5159  
5160  
5161  
5162  
5163  
5164  
5165  
5166  
5167  
5168  
5169  
5170  
5171  
5172  
5173  
5174  
5175  
5176  
5177  
5178  
5179  
5180  
5181  
5182  
5183  
5184

## Chapter 4

# Graph-Based Aggregate Modeling

As we showcase in the last chapter, the relational semantics we are trying to model is challenging to capture in an unsupervised fashion. The information available in each sentence is scarce. To alleviate this problem, we can take a holistic approach by explicitly modeling the relational information at the dataset level, similarly to the aggregate approaches discussed in Section 2.4. The information encoded in the structure of the dataset can be modeled using a graph (Qian et al. 2019). In this chapter, we propose a graph-based unsupervised aggregate relation extraction method to exploit the signal in the dataset structure explicitly.

Since we model dataset-level information, we need to place ourselves in the aggregate setup (Section 2.1) as defined by Equation 2.2. As a reminder, the aggregate setup is in opposition to the sentential setup used in the previous chapter. In the sentential setup, we process sentences independently. In contrast, in the aggregate setup, we consider all the samples  $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{E}$  jointly to extract knowledge base facts  $\mathcal{D}_{\text{KB}} \subseteq \mathcal{E} \times \mathcal{R}$ , without necessarily mapping each individual sample to a fact. We already introduced two aggregate supervised relation extraction approaches relying on graph modeling, label propagation (Section 2.4.1) and EPGNN (Section 2.4.5). The latter uses a spectral graph convolutional network (GCN). GCNs are the main contribution coming from a recent resurgence of interest in graph-based approaches through the use of deep learning methods. It has been shown that these methods share some similarities with the Weisfeiler–Leman isomorphism test (Kipf and Welling 2017). A graph isomorphism test attempts to decide whether two graphs are identical. To this end, it assigns a color to each element, classifying it according to its neighborhood. Coupled with the assumption that sentences conveying similar relations have similar neighborhoods, this closely relates the isomorphism problem to unsupervised relation extraction. However, unsupervised GCNs are usually trained by assuming that neighboring samples have similar representations, completely discarding the characteristic of the Weisfeiler–Leman algorithm that makes it interesting from a relation extraction point of view. In this chapter, we propose alternative training objectives of unsupervised graph neural networks for relation extraction.

In Section 4.1, we see how to extend the definition of a simple graph to model a relation extraction problem. We then provide some statistics on the T-REX dataset in Section 4.2. The results support that large amount of information can be leveraged from topological features for the relation extraction problem. In Section 4.3, we take a quick tour of graph neural

“*C’est même des hypothèses simples qu’il faut le plus se défier, parce que ce sont celles qui ont le plus de chances de passer inaperçues.*

“*It is the simple hypotheses of which one must be most wary; because these are the ones that have the most chances of passing unnoticed.*

— Henri Poincaré, *Thermodynamique* (1908)

“*In an extreme view, the world can be seen as only connections, nothing else. We think of a dictionary as the repository of meaning, but it defines words only in terms of other words. I liked the idea that a piece of information is really defined only by what it’s related to, and how it’s related. There really is little else to meaning. The structure is everything.*

— Tim Berners-Lee, *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor* (1999)

Qian et al., “GraphIE: A Graph-Based Framework for Information Extraction” 2019

Kipf and Welling, “Semi-Supervised Classification with Graph Convolutional Networks” ICLR 2017

5239 networks (GNN) and the Weisfeiler–Leman isomorphism test. Most GNNs  
 5240 apply to simple undirected graphs, whereas we need a more complex struc-  
 5241 ture to encode the relation extraction task. While most models, such as  
 5242 EPGNN, try to adapt the encoding of relation extraction to simple undi-  
 5243 rected graphs, in Section 4.4, we propose to adapt existing GNN methods  
 5244 to the richer structure needed to fully capture the relation extraction prob-  
 5245 lem. Finally, Section 4.5 presents the experimental results of the proposed  
 5246 approaches.

5247

5248 **Notations used in this chapter.** A simple undirected graph is defined  
 5249 as a tuple  $G = (V, E)$  where  $V$  is a set of  $n$  vertices and  $E$  is a set of  $m$   
 5250 edges.<sup>55</sup> An edge  $\{u, v\} \in E$  connects two vertices  $u, v \in V$ , which are then  
 5251 said to be *neighbors*. We use  $N : V \rightarrow 2^V$  to denote the function which  
 5252 associates to each vertex the set of its neighbors  $N(u) = \{v \in V \mid \exists \{u, v\} \in E\}$ .  
 5253 Alternatively, a graph  $G$  can be represented by its adjacency matrix  
 5254  $M \in \{0, 1\}^{n \times n}$ , with  $m_{uv} = 1$  if  $\{u, v\} \in E$  and  $m_{uv} = 0$  otherwise.  
 5255 A graph is said to encode an adjacency relation on its vertices, which  
 5256 foreshadows the remainder of this chapter.

5257

5258

5259

5260

## 4.1 Encoding Relation Extraction as a Graph Problem

5261

5262

5263

5264

5265

5266

5267

5268

5269

5270

5271

5272

5273

5274

5275

5276

5277

5278

5279

5280

5281

5282

5283

5284

5285

5286

5287

5288

5289

5290

5291

5292

In this section, we describe how to frame the relation extraction problem as a problem on graphs. In particular, we describe the structure of an attributed multigraph which is a generalization of the simple undirected graph defined in the previous paragraph. This structure is needed to model entities linked by multiple relations or sentences since this can't readily be done with a simple graph.

Since a knowledge base relation can be formally defined as a set of entity pairs (Section 1.4.1), we can represent it using a single graph  $G = (V, E)$  where  $V$  is the set of entities ( $V = \mathcal{E}$ ) and  $E$  is the set of pairs linked by the relation ( $E \in \mathcal{R}$ ). However, to encode the relation extraction task on a graph, we need different kinds of edges. We, therefore, use the structure of an attributed<sup>56</sup> multigraph  $G = (\mathcal{E}, \mathcal{A}, \varepsilon, \rho, \varsigma)$  where:<sup>57</sup>

- $\mathcal{E}$  is the set of entities, which corresponds to the vertices of  $G$  (indeed  $\mathcal{E} = V$ ),
- $\mathcal{A}$  is the set of arcs, which represent a directed<sup>58</sup> link (usually a sentence) between two entities (this approximately corresponds to the set of edges  $E$  in a simple graph, but can also be seen as equivalent to a supervised set of samples  $\mathcal{D}_{\mathcal{R}}$ ),
- $\varepsilon_1 : \mathcal{A} \rightarrow \mathcal{E}$  assigns to each arc its source vertex (the entity  $e_1$ ),
- $\varepsilon_2 : \mathcal{A} \rightarrow \mathcal{E}$  assigns to each arc its target vertex (the entity  $e_2$ ),
- $\varsigma : \mathcal{A} \rightarrow \mathcal{S}$  assigns to each arc  $a \in \mathcal{A}$  the corresponding sentence containing  $\varepsilon_1(a)$  and  $\varepsilon_2(a)$ ,
- $\rho : \mathcal{A} \rightarrow \mathcal{R}$  assigns to each arc  $a \in \mathcal{A}$  the relation linking the two entities conveyed by  $\varsigma(a)$ .

In this graph, the vertices are entities with an arc linking them for each sentence in which they both appear. Figure 4.1 shows the graph corresponding to the sentences in Table 2.1. Let's call  $a \in \mathcal{A}$  the highlighted bottom left arc in Figure 4.1 linking SMERSH to counterintelligence. Applying the above definitions to this arc we have:

<sup>55</sup> In a simple graph, we always have  $m \leq n(n-1)$  which tightens to  $m \leq n(n-1)/2$  for undirected ones.

The distinction between  $E$  and  $\mathcal{E}$  is important. We decided to keep the usual  $G = (V, E)$  notation for undirected graphs. However, the multigraph we describe in this section has the set of entities  $\mathcal{E}$  as vertices. This set  $\mathcal{E}$  takes the place of  $V$ ; despite the similar notation, it has nothing to do with  $E$ .

<sup>56</sup> The term “*labeled*” is usually reserved for graphs where the domain of attributes is discrete and finite. However the set of possible sentences  $\mathcal{S}$  is not (theoretically) finite.

<sup>57</sup> To be perfectly formal,  $G$  should also depend on  $\mathcal{S}$  and  $\mathcal{R}$ , the codomains of  $\varsigma$  and  $\rho$ . We omit them for conciseness.

<sup>58</sup> We use the word *edge* to refer to a symmetric connection  $\{u, v\}$ , while *arc* refers to an asymmetric connection  $(u, v)$ . Using this nomenclature, an undirected graph has *edges* while a directed graph has *arcs*.

- 5293 •  $\varepsilon_1(a) = \text{SMERSH (Q158363)}$
- 5294 •  $\varepsilon_2(a) = \text{counterintelligence (Q501700)}$
- 5295 •  $\varsigma(a) = \text{In its counter-espionage}_{e_2}$  and counter-intelligence roles,
- 5296  $\text{SMERSH}_{e_1}$  appears to have been extremely successful
- 5297 throughout World War II.
- 5298 •  $\rho(a) = \text{field of work (P101)}$

Remember that  $\mathcal{S}$  is not simply a set of regular sentences but a set of sentences with two tagged and ordered entities.

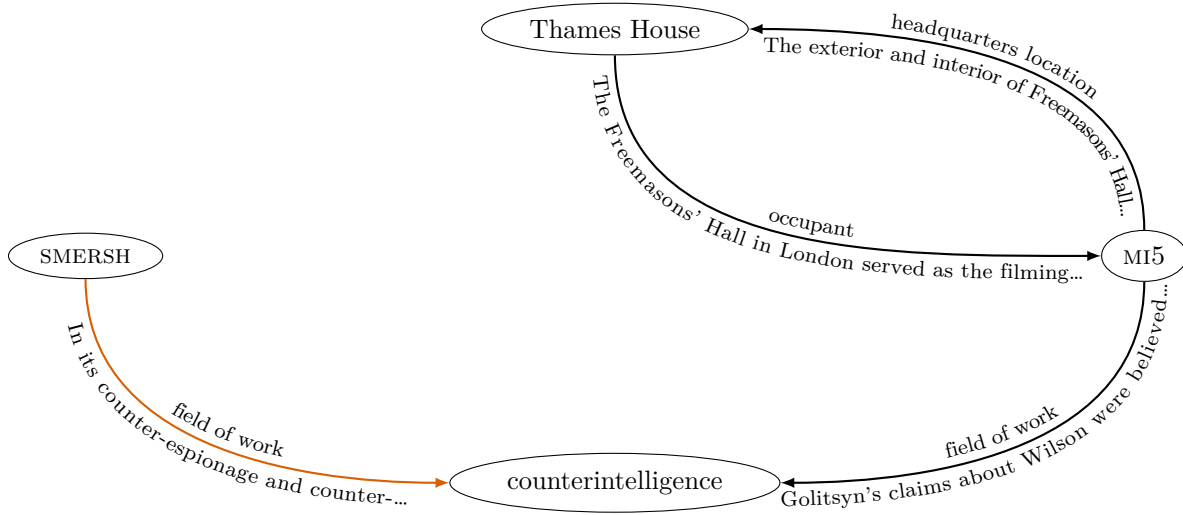


Figure 4.1: Multigraph  $G$  corresponding to the four samples of Table 2.1. For each arc  $a$ , its relation  $\rho(a)$  is written over the arc, and the beginning of the conveying sentence  $\varsigma(a)$  is written under the arc. For ease of reading, surface forms are given instead of numerical identifiers. The highlighted arc corresponds to the example given above.

<sup>59</sup> Indeed, in this case, the graph is simply a set of entities linked by relation arcs such as Sanaa  $\xrightarrow{\text{capital of}}$  Yemen.

<sup>60</sup> For example, while the notion of sink—a vertex with no outgoing arcs—might be of interest to graph theorists, it bears no special meaning in our encoding.

<sup>61</sup> Note that the functions we define here are for the open neighborhood. This means that we don't consider a vertex to be its own neighbor.

In the supervised relation extraction task, the set of relations  $\mathcal{R}$  is fully known, and  $\rho$  is partially known; the goal is to complete  $\rho$ . In the unsupervised relation extraction task,  $\mathcal{R}$  is unknown, and  $\rho$  must be built from the ground up. We can also encode a knowledge base using this structure by removing the associated sentences (i.e. the  $\varsigma$  attributes).<sup>59</sup>

Note that the graph  $G$  is directed because most relations and sentences are asymmetric (inverting the two entities changes the meaning). This is the only semantic associated with orientation.<sup>60</sup> In the unsupervised setting, when the graph is not labeled with relations, each arc  $u \xrightarrow{s} v$  has a symmetric arc  $u \xrightarrow{\tilde{s}} v$  where  $\tilde{s} \in \mathcal{S}$  is the same sentence as  $s \in \mathcal{S}$  with the tags  $\_e_1$  and  $\_e_2$  inverted.

For ease of notation, let us define the incident function  $\mathcal{J}$  associating to each vertex its set of incident arcs  $\mathcal{J}(e) = \{a \in \mathcal{A} \mid \varepsilon_1(a) = e \vee \varepsilon_2(a) = e\}$ . In other words,  $\mathcal{J}$  associates to each entity the set of samples in which it appears. Furthermore, for each relation  $r \in \mathcal{R}$ , we define the relation graphs  $G_{\langle r \rangle} = (\mathcal{E}, \mathcal{A}_{\langle r \rangle}, \varepsilon_1, \varepsilon_2, \rho, \varsigma)$  where  $\mathcal{A}_{\langle r \rangle} = \{a \in \mathcal{A} \mid \rho(a) = r\}$  is the set of arcs labeled with relation  $r$ . We can then define the out-neighbors  $N_{\langle r \rangle}^{\rightarrow}$  and in-neighbors  $N_{\langle r \rangle}^{\leftarrow}$  functions on the relation graph  $G_{\langle r \rangle}$  as follows:<sup>61</sup>

$$N_{\langle r \rangle}^{\rightarrow}(e_1) = \{e_2 \in \mathcal{E} \mid \exists a \in \mathcal{A} : \varepsilon_1(a) = e_1 \wedge \varepsilon_2(a) = e_2 \wedge \rho(a) = r\},$$

$$N_{\langle r \rangle}^{\leftarrow}(e_1) = \{e_2 \in \mathcal{E} \mid \exists a \in \mathcal{A} : \varepsilon_2(a) = e_1 \wedge \varepsilon_1(a) = e_2 \wedge \rho(a) = r\}.$$

Using these definitions we can write expressions for the generic neighbors function:

$$N_{\langle r \rangle}(e) = N_{\langle r \rangle}^{\rightarrow}(e) \cup N_{\langle r \rangle}^{\leftarrow}(e),$$

$$N(e) = \bigcup_{r \in \mathcal{R}} N_{\langle r \rangle}(e).$$

5347 Finally, we can define the degree of a vertex as its number of neighbors:

5348

5349

$$\deg(e) = |N(e)|,$$

5350

5351 which can be broken down into in-degree and out-degree using in-neighbors  
5352 and out-neighbors.

5353

5354 Using these notations we can reformulate modeling assumptions such  
5355 as  $\mathcal{H}_{\text{BICLIQUE}}$  (Section 2.5.4),  $\mathcal{H}_{1\text{-ADJACENCY}}$  (Section 2.3.2) and  $\mathcal{H}_{1 \rightarrow 1}$  (Sec-  
5356 tion 2.5.6). For example, the hypothesis  $\mathcal{H}_{\text{BICLIQUE}}$  draw its name from the  
5357 fact that for all relation  $r \in \mathcal{R}$ , the relation graph  $G_{(r)}$  is assumed to be  
5358 a biclique.<sup>62</sup> This is especially of interest to study matching the blanks  
(MTB, Section 2.5.6). It can be analyzed using the following graph:

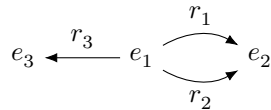
5359

5360

5361

5362

5363



5364

5365 MTB makes two main assumptions:  $\mathcal{H}_{1\text{-ADJACENCY}}$  and  $\mathcal{H}_{1 \rightarrow 1}$ . In the above  
5366 graph,  $\mathcal{H}_{1\text{-ADJACENCY}}$  implies that  $r_1$  and  $r_2$  should be the same, while  $\mathcal{H}_{1 \rightarrow 1}$   
5367 implies that  $r_3$  should be different from  $r_1$  and  $r_2$ . From this simple exam-  
5368 ple, we can also see that MTB training is 1-localized, which means that it  
5369 only exploits the fact that two samples are direct neighbors.<sup>63</sup> In contrast,  
5370 a sentential approach is 0-localized; it completely ignores other samples.  
5371 This is actually the case of MTB during evaluation. The same problem  
5372 plagues the fill-in-the-blank model of Chapter 3; while training is influ-  
5373 enced by the direct neighbors (through the entity embeddings), when  
5374 classifying an unknown sample, its neighbors are ignored. The goal of this  
5375 chapter is to consider larger neighborhoods both for training unsupervised  
5376 models and for making predictions with them.

5376

5377

5378

## 4.2 Preliminary Analysis and Proof of Principle

5379

5380

5381

5382

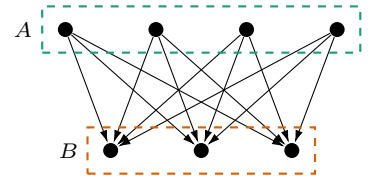
5383 In this section, we want to ensure the soundness of graph-based approaches  
5384 by providing some statistics about a large relation extraction dataset. In  
5385 particular, we start by building an attributed multigraph as described  
5386 in Section 4.1. We focus on T-REX (Section C.7, Elshahar et al. 2018),  
5387 an alignment (Section 2.2.2) of Wikipedia with Wikidata. This dataset  
5388 has the advantage of being both large and publicly available. Note that  
5389 the graph we analyze in this section is not a knowledge base. Each arc  
5390 is both labeled with a relation and attributed with a sentence. The fact  
5391 that several arcs are incident to a vertex does not necessarily imply that  
5392 the corresponding entity is linked by several relations, only that it was  
5393 mentioned multiple times.

5393

5394 Figure 4.2 shows the distribution of vertices’ degrees in the graph as-  
5395 sociated with T-REX. The first thing we can notice about this graph is  
5396 that it is *scale-free*. This means that a random vertex  $v \in \mathcal{E}$  has degree  
5397  $\deg(v) = k$  with probability  $P(k) \propto k^{-\gamma}$  for a parameter  $\gamma$  which depends  
5398 on the graph. In other words, the distribution of degrees follows a power  
5399 law. In a scale-free graph, a lot of vertices have few neighbors. In contrast,  
5400 the distribution of degrees in a random Erdős–Rényi graph<sup>64</sup> is expected  
to follow a binomial distribution. Scale-free graphs occur in a number of

Since we mention several hypotheses, we take this opportunity to remind the reader that all assumptions are detailed in Appendix B.

<sup>62</sup> A biclique is a *complete bipartite graph*. Its vertices can be split into two sets  $A, B \subseteq \mathcal{E}$  such that each vertex in  $A$  is linked to all vertices in  $B$ . For example:



<sup>63</sup> Here we use *neighbors* as in “arc-neighbors.” This is a relation between two arcs sharing a common endpoint. Arc-neighbors are simple neighbors in the line graph described in Section 4.4.1.

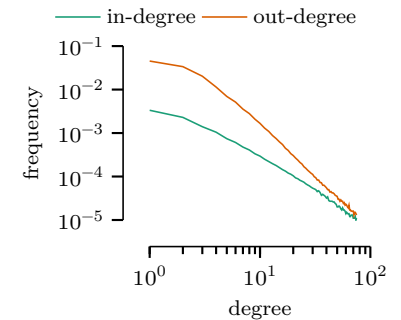


Figure 4.2: T-REX vertices degree distribution. The lines give the frequency of vertices with the given in- and out-degree in the dataset. Note that both axes are log-scaled. This plot was cut at a degree of 75, which corresponds to a minimum frequency of  $10^{-5}$  out of a total of 19 392 185 arcs. In reality, the vertex with the maximum degree is “United States of America” Q30 with an in-degree of 1 522 224. The asymmetry between the distribution of in-degrees and out-degrees can be explained by the fact that knowledge bases prefer to encode many-to-one relations instead of their one-to-many converse.

<sup>64</sup> There are several different ways to sample random graphs; the Erdős–Rényi model is one of them. In this model, arcs are incrementally added between two uniformly chosen vertices. In contrast, if vertices with already high degrees are selected more often (the Barabási–Albert model), the resulting graph is scale-free.

5401 contexts, such as social networks and graphs of linked web pages. Most  
 5402 unsupervised relation extraction datasets and knowledge bases should be  
 5403 expected to be scale-free. This needs to be kept in mind when designing  
 5404 graph-processing algorithms for relation extraction. Indeed most vertices  
 5405 have a small neighborhood, so we might be tempted to take neighbors of  
 5406 neighbors carelessly. However, scale-free graphs have a very small diame-  
 5407 ter<sup>65</sup>  $D \in O(\log \log n)$ . This means that we can quickly reach most vertices  
 5408 following a small number of arcs. This is in part due to the fact that some  
 5409 vertices have very high degree, for example in T-REx, the vertex “United  
 5410 States of America” Q30 is highly connected with  $\text{deg}(\text{Q30}) = 1\,697\,334$ .  
 5411 In particular, this implies that by considering neighbors of neighbors, we  
 5412 quickly need to consider the whole graph; this is particularly problematic  
 5413 for graph convolutional networks described in Section 4.3.

5414 We now come to the main incentive for taking a graph-based approach  
 5415 to the unsupervised relation extraction task:

5416 **Hypothesis:** *In the relation extraction problem, we can get additional*  
 5417 *information from the neighborhood of a sample.*  
 5418

5419 To test this hypothesis, we compute statistics on the distribution of neigh-  
 5420 bors. However, as we just saw, the support of this distribution is of high  
 5421 dimension. Hence, we look at the statistics of paths in our multigraph.<sup>66</sup>  
 5422 As a graph theory reminder, we can formally define a path as follows:

- 5423 • A *walk* on length  $n$  is a sequence of arcs  $a_1, a_2, \dots, a_n \in \mathcal{A}$  such that  
 5424  $\varepsilon_2(a_{i-1}) = \varepsilon_1(a_i)$  for all  $i = 2, \dots, n$ .
- 5425 • A *trail* is a walk with  $a_i \neq a_j$  for all  $1 \leq i < j \leq n$  (arcs do not  
 5426 repeat). In practice this means that  $(s, e)$  do not repeat. It is not  
 5427 a statement about relations conveyed by these arcs; it is entirely  
 5428 possible that for some  $i, j$  we have  $\rho(a_i) = \rho(a_j)$ .
- 5429 • A *path* is a trail with  $\varepsilon_1(a_i) \neq \varepsilon_1(a_j)$  for all  $1 \leq i < j \leq n$  (vertices  
 5430 do not repeat).

5431 It is also possible to base these definitions on *open walks*, which are walks  
 5432 where  $\varepsilon_1(a_1) \neq \varepsilon_2(a_n)$  (the walk does not end where it started). We base  
 5433 the discussion of this section around the following random path:  
 5434

$$5435 \quad e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3 \xrightarrow{r_3} e_4,$$

5436  
 5437 Using these definitions, we can restate our hypothesis. In this path, we  
 5438 expect  $r_2 \not\perp r_1$  and  $r_2 \not\perp r_3$ . However, enumerating all possible paths in a  
 5439 graph with  $n = 2\,819\,966$  vertices and  $m = 19\,392\,185$  arcs is not practical.

5440 To approximate path statistics, we turn to sampling. However, uni-  
 5441 formly sampling paths is not straightforward. As a first intuition, to uni-  
 5442 formly sample a path of length 1—that is, an arc—we can use the following  
 5443 procedure:  
 5444

- 5445 1. Sample an entity  $e_1$  weighted by its degree,  
 5446  $e_1 \sim \text{Cat}(\mathcal{E}, e \mapsto \text{deg}(e) / 2m)$
- 5447 2. Uniformly sample an arc incident to the entity  $e_1$ .  
 5448  $a \sim \mathcal{U}(\mathcal{J}(e_1))$

5449 The first vertex we select must be weighted by how many paths start  
 5450 there, and since paths of length 1 are arcs, we weight each vertex by its  
 5451 degree.<sup>67</sup> If we want to sample paths of length 2, the first vertex must be  
 5452 selected according to the number of paths of length 2 starting there. Then  
 5453 the second vertex is selected among the neighbors of the first weighted by  
 5454 the number of paths of length 1 starting there, etc.

<sup>65</sup> The diameter of a graph is the length of the longest shortest-path:

$$D = \max_{u,v \in \mathcal{E}} \delta(u, v),$$

where  $\delta(u, v)$  is the length of the short-  
 est path from  $u$  to  $v$ .

<sup>66</sup> Paths of length  $k$  are in a domain of size  $|\mathcal{R}|^k$ , whereas neighbors are in a domain of size  $|\mathcal{R}|^{\Delta(G)}$  with  $\Delta(G)$  designating the maximum degree in  $G$ . By studying paths of length 3, we are effectively studying a subsampled neighborhood of the central arc.

The symbol  $\not\perp$  is used to mean “not independent”:

$$a \not\perp b \iff P(a, b) \neq P(a)P(b)$$

$\text{Cat}(\mathcal{E}, f)$  refers to the Categorical distribution over the set  $\mathcal{E}$  where the probability of picking  $e \in \mathcal{E}$  is  $f(e)$ . The  $2m$  appears from the normalization factor  $\sum_{e \in \mathcal{E}} \text{deg}(e) = 2m$ .

<sup>67</sup> To give an intuition, we can also think of what would happen if we chose both the entity and incident arc uniformly. An arc that links two entities otherwise unrelated to any other entities is likely to be sampled since sampling any of its two endpoints as  $e_1$  would guarantee we select this arc. On

5455 **algorithm** PATH COUNTING  
5456 *Inputs:*  $G = (\mathcal{E}, \mathcal{A}, \varepsilon, \rho, \varsigma)$  relation multigraph  
5457  $k$  paths length  
5458 *Output:*  $C$  relation paths counter  
5459  $\triangleright$  Initialization  $\triangleleft$   
5460  $C \leftarrow$  new counter  $\mathcal{X}^k \rightarrow \mathbb{R}$  initialized at 0  
5461  $\triangleright$  Main Loop  $\triangleleft$   
5462 **loop**  
5463  $\triangleright$  Initialize the importance weight with  $\mathcal{W}^k$   $\triangleleft$   
5464  $w \leftarrow (\mathbf{1}^\top \mathbf{M}^k \mathbf{1})^{-1}$   $\triangleright$   $\mathbf{M}$  is the adjacency matrix  
5465 Initialize empty walk  $\mathbf{a} = ()$   
5466 Sample  $v \sim \mathcal{U}(\mathcal{E})$   
5467  $w \leftarrow n \times w$   $\triangleright$  Update  $w$  following the sampling of  $v$   
5468 **for**  $i = 1, \dots, k$  **do**  
5469 Sample  $x \sim \mathcal{U}(\mathcal{J}(v))$   
5470  $w \leftarrow w \times \text{deg}(v)$   $\triangleright$  Accumulate  $1 / \mathcal{F}^k$   
5471 **if**  $\varepsilon_1(x) = v$  **then**  $\triangleright$  Continue with  $\varepsilon(x) \setminus \{v\}$   
5472 Append  $x$  to  $\mathbf{a}$   
5473  $v \leftarrow \varepsilon_2(x)$   
5474 **else**  
5475 Append  $\tilde{x}$  to  $\mathbf{a}$   
5476  $v \leftarrow \varepsilon_1(x)$   
5477 **if**  $\mathbf{a}$  is a path **then**  
5478  $\mathbf{r} \leftarrow (\rho(a_i))_{1 \leq i \leq k}$   $\triangleright$  Take the relations of  $\mathbf{a}$   
5479  $C[\mathbf{r}] \leftarrow C[\mathbf{r}] + w$   
5480 **output**  $C$   
5481

the other hand, an arc whose both endpoints have high degrees has little chance of being sampled since even if one of its endpoints is selected as  $e_1$  in the first step, the arc is unlikely to be selected in the second step.

Algorithm 4.1: Path counting algorithm. The higher the number of iterations of the main loop, the more precise the results will be. In our experiments, we used one billion iterations. The inner for loop builds the walk  $\mathbf{a}$ . If it is a correct path, the relation type of the path is added to the counter with importance weight  $w$ . For numerical stability, we actually compute  $w$  in log-space. The initial factor  $n = |\mathcal{E}|$  in  $w$  comes from the preceding uniform sampling of  $v$  from  $\mathcal{E}$ , which is part of the computation of  $\mathcal{F}^k$ .

5482 Sadly enough, counting paths is #P-complete<sup>68</sup> (Valiant 1979) so we  
5483 must rely on the regularity of our graph and turn to approximate algo-  
5484 rithms. We propose to use the number of walks as an approximation of  
5485 the number of paths.<sup>69</sup> A classical result on simple graphs  $G = (V, E)$  is  
5486 that the powers of the adjacency matrix  $\mathbf{M}$  count the number of walks  
5487 between pairs of vertices. For any two vertices  $u, v \in V$ , the value  $m_{uv}^k$ —to  
5488 be interpreted as  $(\mathbf{M}^k)_{uv}$ —is the number of walks of length  $k$  from  $u$  to  
5489  $v$ . In the case of our multigraph, if we wish to count walks, the adjacency  
5490 matrix should contain the number of arcs—that is, the number of walks  
5491 of length 1—between vertices.

5492 We could then build a Monte Carlo estimate by following the naive  
5493 procedure above of sampling vertices one by one according to the number  
5494 of walks starting with them. Let's call  $\mathcal{W}^k$  this distribution over walks of  
5495 length  $k$ . Sampling from  $\mathcal{W}^k$  is particularly slow since it involves sampling  
5496 from a categorical distribution over thousands of elements. Since we only  
5497 want to evaluate a (counting) function over an expectation  $\mathbb{E}_{\mathbf{a} \sim \mathcal{W}^k}$ , we can  
5498 instead perform importance sampling. We use the substitute distribution  
5499  $\mathcal{F}^k$  that uniformly selects a random neighbor at each step. To make this  
5500 trick work, we only need to compute the importance weights  $\frac{\mathcal{W}^k(\mathbf{a})}{\mathcal{F}^k(\mathbf{a})}$  for all  
5501 walks  $\mathbf{a} \in \mathcal{A}^k$ . Since  $\mathcal{W}^k$  is the uniform distribution over all walks, it is  
5502 constant  $\mathcal{W}^k(\mathbf{a}) = (\mathbf{1}^\top \mathbf{M}^k \mathbf{1})^{-1}$ . On the other hand  $\mathcal{F}^k(\mathbf{a})$  can be trivially  
5503 computed as the product of inverse degrees of  $a_i$ . The resulting counting  
5504 procedure is listed as Algorithm 4.1. We still need to reject non-paths at  
5505 the end of the main loop. Note that this algorithm is not exact since the  
5506 importance weights  $w$  are computed from the number of walks, not paths.  
5507

5508 Using this algorithm on one billion samples from T-REx, we find that

<sup>68</sup> A functional complexity class at least as hard as NP-complete.

<sup>69</sup> Other approximations of path counting exist (Roberts and Kroese 2007), but the approach we propose is particularly suited to our multigraph. In particular, the shape parameter  $\gamma$  of our degree distribution is relatively small, which produces a large number of outliers. Our importance-sampling-based approach allows us to reduce the variance of the frequency estimations.



Frequency	Relation path	
	Surface forms	Identifiers
54.657%	<i>country • diplomatic relation • country</i>	P17 • P530 • P17
31.696%	<i>country • diplomatic relation • citizen of</i>	P17 • P530 • P27
6.680%	<i>country • shares border with • citizen of</i>	P17 • P47 • P27
0.013%	<i>country • seceded from • citizen of</i>	P17 • P807 • P27
9.445%	<i>sport • sport • member of<sub>ST</sub></i>	P641 • P641 • P54
10 <sup>-6</sup> %	<i>sport • industry • member of<sub>ST</sub></i>	P641 • P452 • P54

Table 4.1: Frequencies of some paths of length 3 in T-REX. The first column gives the approximate per mille frequency of paths with the given type. It is computed as the importance weight attributed to the path by the counter  $C$  in Algorithm 4.1 divided by the sum of all importance weights in  $C$ . We use  $_{ST}$  as an abbreviation of “sport team.” The path in the first row is the most frequent one in the dataset; other paths were selected for illustrative purposes. The last path was sampled a single time with an importance weight of 0.89.

<sup>70</sup> This is not surprising as most general knowledge datasets are dominated by geopolitical entities and relations.

<sup>71</sup> This is not a conditional entropy. The context relations  $r_1, r_3$  are fixed; they correspond to elementary events, not random variables (as shown by the fact that they are italicized, not up-shape).

As a reference for the remainder of this section, the distribution of relation in T-REX has an entropy of  $H(r) \approx 6.26$  bits. This is for a domain of  $|\mathcal{R}| = 1\,316$  relations.

To give a first intuition of what this value represents, we take once again the trivial example of  $r_1 = \textit{sport}$  and  $r_3 = \textit{member of}_{ST}$ . In this case,  $D_{KL}(P(r_2 | r_1, r_3) \| P(r_2)) \approx 5.47$  bits. This is due to the fact that encoding  $r_2$  given its neighbors necessitates close to 0 bits (as shown in Table 4.1,  $r_2$  almost always takes the value  $\textit{sport}$ ) but encoding  $\textit{sport}$  among all possible relations in  $\mathcal{R}$  necessitates 5.47 bits (which is a bit less than most relations since  $\textit{sport}$  commonly appears in T-REX).

<sup>72</sup> We denote the cross-entropy by  $H_Q(P) = -\mathbb{E}_P[\log Q]$ .

the most common paths of length three are related to geopolitical relations,<sup>70</sup> see Table 4.1. Let us now turn to statistics that could help relation extraction models. To showcase the dependency between a sample’s relation  $r_2$  and its neighbors  $r_1$  and  $r_3$ , we investigate the distribution  $P(r_2 | r_1, r_3)$ . In other words, given a sample, we want to see how its relation is influenced by the relations of two neighboring samples.

The first value we can look at is the entropy<sup>71</sup>  $H(r_2 | r_1, r_3)$ . For example, in the case of  $r_1 = \textit{sport}$  and  $r_3 = \textit{member of}_{ST}$ , all observed values of  $r_2$  are given in Table 4.1. All of them were  $\textit{sport}$  with the exception of a single path, which means that  $H(r_2 | r_1, r_3) \approx 0$ . In other words, if we are given a sample  $(s, e) \in \mathcal{D}$  and we suspect another sentence containing  $e_1$  to convey  $\textit{sport}$  and another containing  $e_2$  to convey  $\textit{member of}_{ST}$ , we can be almost certain that the sample  $(s, e)$  conveys  $\textit{sport}$ .

To measure this type of dependency at the level of the dataset, we can look at the following value:

$$D_{KL}(P(r_2 | r_1, r_3) \| P(r_2))$$

The Kullback–Leibler divergence is also called the *relative entropy*. Indeed,  $D_{KL}(P \| Q)$  can be interpreted as the additional quantity of information needed to encode  $P$  using the (suboptimal) entropy encoding given by  $Q$ . If this value is 0, it means that no additional information was provided by  $r_1$  and  $r_3$ . When marginalizing over all possible contexts  $r_1, r_3$ , we obtain the mutual information between the relation of a sample  $r_2$  and the relation of two of its neighbors. On T-REX, we observe:

$$I(r_2; r_1, r_3) \approx 6.95 \text{ bits}$$

In other words, we can gain 6.95 bits of information simply by modeling two neighbors (one per entity). These 6.95 bits can be interpreted as the number of bits needed to perfectly encode  $r_2$  given  $r_1, r_3$  (the conditional entropy  $H(r_2 | r_1, r_3) \approx 1.06$  bits) subtracted from the number of bits needed to encode  $r_2$  without looking at its neighbors (the cross-entropy  $\mathbb{E}_{r_1, r_3}[H_{P(r_2)}(r_2 | r_1, r_3)] \approx 8.01$  bits).<sup>72</sup> In other words, most of the uncertainty about the relation of a sample can be removed by looking at the relations of two of its neighbors.

## 4.3 Related Work

In the previous section, we show that the attributed multigraph encoding we introduced in Section 4.1 can help us leverage additional information for the relation extraction task. In this section, we present the existing

5563 framework for computing distributed representations of graphs. In most  
 5564 cases, these process simple undirected graphs  $G = (V, E)$ . Still, these  
 5565 methods are applicable to our relation extraction multigraph with some  
 5566 modifications, as shown in Sections 4.3.4 and 4.4.

5567 The use of graphs in deep learning has seen a recent surge of interest  
 5568 over the last few years. This produced a set of models known as graph  
 5569 neural networks (GNN) and graph convolutional networks (GCN).<sup>73</sup> While  
 5570 the first works on GNN started more than twenty years ago (Sperduti  
 5571 and Starita 1997), we won't go into a detailed historical review, and we  
 5572 exclusively focus on recent models. Note that we already presented an older  
 5573 graph-based approach in Section 2.4.1, the label propagation algorithm.  
 5574 We also discussed EPGNN in Section 2.4.5, which is a model built on top  
 5575 of a GCN. We further draw parallels between EPGNN and our proposed  
 5576 approach in Section 4.4.1.

5577 The thread of reasoning behind this section is as follows:

- 5578 • We present the “usual” way to process graphs (Sections 4.3.1–4.3.4).
- 5579 • We present the theory behind these methods (Section 4.3.5).
- 5580 • We show how this theoretical background can help us design a new  
 5581 approach specific to the unsupervised relation extraction task (Sec-  
 5582 tion 4.4).

5583 In this related work overview, we mainly describe algorithms working on  
 5584 standard  $G = (V, E)$  graphs, not the labeled multigraphs of Section 4.1,  
 5585 with the exception of Section 4.3.4. We start by quickly describing models  
 5586 based on random walks in Section 4.3.1; these are spatial methods which  
 5587 serve as a gentle introduction to the manipulation of graphs by neural  
 5588 networks. Furthermore, they were influential in the development of sub-  
 5589 sequent models and in our preliminary analysis with computation of path  
 5590 statistics (Section 4.2), which allows us to draw parallels with more mod-  
 5591 ern approaches. We then introduce the two main classes of GCN—which  
 5592 consequently are also the two main classes of GNN—used nowadays: spec-  
 5593 tral (Section 4.3.2) and spatial (Section 4.3.3). Apart from the few works  
 5594 mentioned in Chapter 2, GNNs were seldom used for relation extraction.  
 5595 We, therefore, focus on the evaluation of GNN on an entity classification  
 5596 task, which while different from our problem, works on similar data. In  
 5597 Section 4.3.4, we describe models designed to handle relational data in a  
 5598 knowledge base, in particular R-GCN. We close this related work with a  
 5599 presentation of the Weisfeiler–Leman isomorphism test in Section 4.3.5;  
 5600 it serves as a theoretical motivation behind both GCNs and our proposed  
 5601 approach.

5602

5603

### 5604 4.3.1 Random-Walk-Based Models

5605

5606 DeepWalk (Perozzi et al. 2014) is a method to learn vertex representa-  
 5607 tions from the structure of the graph alone. The representations encode  
 5608 how likely it is for two vertices to be close to each other in the graph. To  
 5609 this end, DeepWalk models the likelihood of random walks in the graph  
 5610 (Section 4.2). These walks are simply sequences of vertices. To obtain a  
 5611 distributed representation out of them, we can use the NLP approaches  
 5612 of Sections 1.2 and 1.3 by treating the set of vertices as the vocabulary  
 5613  $V = \mathcal{E}$ . In particular, DeepWalk uses the skip-gram model of Word2vec  
 5614 (Section 1.2.1.1), using hierarchical softmax to approximate the partition  
 5615 function over all words—i.e. vertices. Vertices part of the same random  
 5616 walk are used as positive examples. In the same way that learning rep-

<sup>73</sup> The term GCN is used with different meanings by various authors. GCNs are always GNNs, but the reverse is not true. However, in practice, the GNNs we describe in this section can essentially be described as GCNs. We use the term GCN to describe models whose purpose is to have a similar function on graphs as CNNs have on images. Some authors only refer to the model of Kipf and Welling (2017) described in Section 4.3.2 as a GCN. In this case, what we call GCN can be called convGNN (convolutional graph neural networks). In any case, GNN and GCN can be considered almost synonymous for the purpose of this thesis since we don't describe any exotic GNN which clearly falls outside of the realm of GCN.

Perozzi et al., “DeepWalk: Online Learning of Social Representations” KDD 2014

5617 resentations to predict the neighborhood of a word gives good word rep-  
 5618 resentations, modeling the neighborhood of a vertex gives good vertex  
 5619 representations.

5620 Perozzi et al. (2014) evaluate their model on a node classification task.  
 5621 For example, one of the datasets they use is BlogCatalog (Tang and Liu  
 5622 2009), where vertices correspond to blogs, edges are built from social net-  
 5623 work connections between the various bloggers, and predicted labels are  
 5624 the set of topics on which each blog focuses. DeepWalk is a transduc-  
 5625 tive method but was extended into an inductive approach called planetoid  
 5626 (Yang et al. 2016). Planetoid also proposes an evaluation on an entity  
 5627 classification task performed on the NELL dataset. The goal of this task  
 5628 is to find the type of an entity (e.g. person, organization, location...) in  
 5629 a knowledge base (Section 1.4). To this end, a special bipartite<sup>74</sup> graph  
 5630  $G_B = (V_B, E_B)$  is constructed where  $V_B = \mathcal{E} \cup \mathcal{R}$  and:

$$5631 E_B = \{ \{e, r\} \subseteq V_B \mid \exists e' \in \mathcal{E} : (e, r, e') \in \mathcal{D}_{KB} \vee (e', r, e) \in \mathcal{D}_{KB} \}$$

5632 This clearly assumes  $\mathcal{H}_{\text{BI CLIQUE}}$ : for each relation the information of “which  
 5633  $e_1$ ” corresponds to “which  $e_2$ ” is discarded. However this information is  
 5634 not as crucial for entity classification as it is for relation extraction. A  
 5635 small example of graph  $G_B$  obtained this way is given in Figure 4.3. The  
 5636 model is trained by jointly optimizing the negative sampling loss and the  
 5637 the log-likelihood of labeled examples. On unseen entities, planetoid reach  
 5638 an accuracy of 61.9% when only 0.1% of entities are labeled.

5639 Using random walks allows DeepWalk and planetoid to leverage the  
 5640 pre-existing NLP literature. However, for each sample, only a small frac-  
 5641 tion of the neighborhood—two neighbors at most—of each node is consid-  
 5642 ered to make a prediction. Subsequent methods focused on modeling the  
 5643 information of the whole neighborhood jointly.

### 5644 4.3.2 Spectral GCN

5645 The first approaches to successfully model the neighborhood of vertices  
 5646 jointly were based on spectral graph theory (Bruna et al. 2014). In practice,  
 5647 this means that the graph is manipulated through its Laplacian matrix  
 5648 instead of directly through the adjacency matrix. In this section, we base  
 5649 our presentation of spectral methods on the work of Kipf and Welling  
 5650 (2017).

5651 We start by introducing some basic concepts from spectral graph theory  
 5652 used to define the convolution operator on graphs. The Laplacian of  
 5653 an undirected graph  $G = (V, E)$  can be defined as:

$$5654 L_C = D - M, \quad (4.1)$$

5655 where  $D \in \mathbb{R}^{n \times n}$  is the diagonal matrix of vertex degrees  $d_{ii} = \text{deg}(v_i)$  and  
 5656  $M \in \mathbb{R}^{n \times n}$  is the adjacency matrix. Equation 4.1 defines the combinatorial  
 5657 Laplacian; however, spectral GCNs are usually defined on the normalized  
 5658 symmetric Laplacian:

$$5659 L_{\text{SYM}} = D^{-1/2} L_C D^{-1/2} = I - D^{-1/2} M D^{-1/2}.$$

5660 Using this definition, we can then take the eigendecomposition of the  
 5661 Laplacian  $L_{\text{SYM}} = U \Lambda U^{-1}$ , where  $\Lambda$  is the ordered spectrum—the diag-  
 5662 onal matrix of eigenvalues sorted in increasing order—and  $U$  is the matrix

Yang et al., “Revisiting Semi-Supervised Learning with Graph Embeddings” ICML 2016

74 A bipartite graph is a graph  $G = (V, E)$  where the vertices can be split into two disjoint sets  $V_1 \cup V_2 = V$  such that all edges  $e \in E$  have one endpoint in  $V_1$  and one endpoint in  $V_2$ .

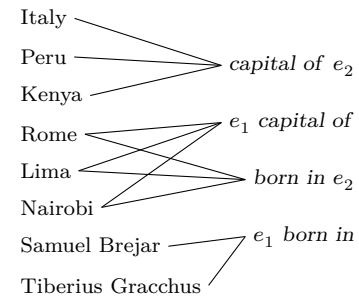


Figure 4.3: NELL dataset bipartite graph. Entities are on the left, while relation slots are on the right. In this graph, the edges are left unlabeled.

Kipf and Welling, “Semi-Supervised Classification with Graph Convolutional Networks” ICLR 2017

The graph Laplacian is similar to the standard Laplacian measuring the divergence of the gradient ( $\Delta = \nabla^2$ ) of scalar functions. Except that the graph gradient is an operator mapping a function on vertices to a function on edges:

$$(\nabla \mathbf{f})_{ij} = f_i - f_j$$

And that the graph divergence is an operator mapping a function on edges to a function on vertices:

$$(\text{div } \mathbf{G})_i = \sum_{j \in V} m_{ij} g_{ij}$$

Given these definitions, the graph Laplacian is defined as  $\Delta = -\text{div } \nabla$ . Applying  $\Delta$  to a signal  $\mathbf{x} \in \mathbb{R}^n$  is equivalent to multiplying this signal by  $L_C$  as defined in Equation 4.1:  $\Delta \mathbf{x} = L_C \mathbf{x}$ .

5671 of normalized eigenvectors. For an undirected graph, the matrix  $\mathbf{M}$  is sym-  
5672 metric, therefore  $\mathbf{U}$  is orthogonal. The orthonormal space formed by the  
5673 normalized eigenvectors is the Fourier space of the graph. In other words,  
5674 we can define the graph Fourier transform of a signal  $\mathbf{x} \in \mathbb{R}^V$  as:

$$5675 \mathcal{F}(\mathbf{x}) = \mathbf{U}^\top \mathbf{x}. \quad 5676$$

5677 Furthermore since the induced space is orthogonal, the inverse Fourier  
5678 transform is simply defined as:

$$5680 \mathcal{F}^{-1}(\mathbf{x}) = \mathbf{U} \mathbf{x}. \quad 5681$$

5682 Having defined the Fourier transform on graphs, we can use the defini-  
5683 tion of convolutions as multiplications in the Fourier domain to define  
5684 convolution on graphs:

$$5686 \mathbf{x} * \mathbf{w} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}(\mathbf{w})), \quad (4.2) \quad 5687$$

5688 where  $\odot$  denotes the Hadamard (element-wise) product. Note that the  
5689 convolution operator implicitly depends on the graph  $G$  since  $\mathbf{U}$  is defined  
5690 from the adjacency matrix  $\mathbf{M}$ . The signal  $\mathbf{w}$  in Equation 4.2 has the  
5691 same function as the parametrized filter of CNN (Equation 1.7). Instead of  
5692 learning  $\mathbf{w}$  in the spatial domain, we can directly parametrize its Fourier  
5693 transform  $\mathbf{w}_\theta = \text{diag}(\mathcal{F}(\mathbf{w}))$ , simplifying Equation 4.2 into:

$$5694 \mathbf{x} * \mathbf{w}_\theta = \mathbf{U} \mathbf{w}_\theta \mathbf{U}^\top \mathbf{x}. \quad (4.3) \quad 5695$$

5696 While  $\mathbf{w}_\theta$  could be learned directly (Bruna et al. 2014), Defferrard et al.  
5697 (2016) propose to approximate it by Chebyshev polynomials of the first  
5698 kind ( $T_k$ ) of the spectrum  $\mathbf{A}$ :

$$5700 \mathbf{w}_\theta(\mathbf{A}) = \sum_{k=0}^K \theta_k T_k(\mathbf{A}). \quad (4.4) \quad 5701$$

5702 The rationale is that computing the eigendecomposition of the graph  
5703 Laplacian is too computationally expensive. The Chebyshev polynomi-  
5704 als approximation is used to localize the filter; since the  $k$ -th Chebyshev  
5705 polynomial is of degree  $k$ , only values of vertices at a distance of at most  
5706  $k$  are needed.<sup>75</sup> This is similar to how CNNs are usually computed; simple  
5707 very localized filters are used instead of taking the Fourier transform of  
5708 the whole input matrix to compute convolution with arbitrarily complex  
5709 functions. Chebyshev polynomials of the first kind are defined as:

$$5710 T_k(\cos x) = \cos(kx). \quad (4.5) \quad 5711$$

5712 They form a sequence of orthogonal polynomials on the interval  $[-1, 1]$   
5713 with respect to the weight  $1 / \sqrt{1 - x^2}$ , meaning that for  $k \neq k'$ :

$$5714 \int_{-1}^1 T_k(x) T_{k'}(x) \frac{dx}{\sqrt{1 - x^2}} = 0. \quad 5715$$

5716 The filter defined by Equation 4.4 is  $K$ -localized, meaning that the  
5717 value of the output signal on a vertex  $v$  is computed from the value of  
5718  $\mathbf{x}$  on vertices at distance at most  $K$  of  $v$ . This can be seen by plugging  
5719 Equation 4.4 back into Equation 4.3, noticing that it depends on the  $k$ -th  
5720 power of the Laplacian and thus of the adjacency matrix.<sup>76</sup>  
5721  
5722  
5723  
5724

The expansion of signals in terms of eigenfunctions of the Laplace operator is the leading parallel between the graph Fourier transform and the classical Fourier transform on  $\mathbb{R}$  (Shuman et al. 2013). In  $\mathbb{R}$ , the eigenfunctions  $\xi \mapsto e^{2\pi i \xi x}$  correspond to low frequencies when  $x$  is small. In the same way, the eigenvectors of the graph Laplacian associated with small eigenvalues assign similar values to neighboring vertices. In particular the eigenvector associated with the eigenvalue 0 is constant with value  $1 / \sqrt{n}$ . On the other hand, eigenvectors associated with large eigenvalues correspond to high frequencies and encode larger changes of value between neighboring vertices.

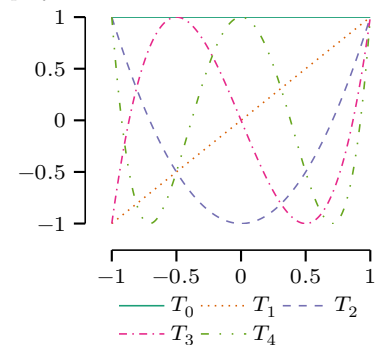
$\text{diag}(\mathbf{x})$  is the diagonal matrix with values of the vector  $\mathbf{x}$  along its diagonal.

<sup>75</sup> The reasoning behind this localization is the same as the one underlying the fact that the  $k$ -th power of the adjacency matrix gives the number of walks of length  $k$  (Section 4.2).

Despite its appearance, Equation 4.5 defines a series of polynomials which can be obtained through the application of various trigonometric identities. An alternative but equivalent definition is through the following recursion:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) \end{aligned}$$

The plot of the first five Chebyshev polynomials of the first kind follows:



5725 Kipf and Welling (2017) proposed to use  $K = 1$  with several further  
 5726 optimizations we won't delve into. Using  $K = 1$  means that their method  
 5727 computes the activation of a node only from its activation and the activa-  
 5728 tions of its neighbors at the previous layer. This makes the GCN of Kipf and  
 5729 Welling (2017) quite similar to spatial methods described in Section 4.3.3.  
 5730 All the equations given thus far were for a single scalar signal; however,  
 5731 we usually work with vector representations for all nodes,  $\mathbf{X} \in \mathbb{R}^{n \times d}$ . In  
 5732 this case, the layer  $\ell$  of a GCN can be described as:

$$5733 \mathbf{H}^{(\ell+1)} = \text{ReLU} \left( (\mathbf{D} + \mathbf{I})^{-1/2} (\mathbf{M} + \mathbf{I}) (\mathbf{D} + \mathbf{I})^{-1/2} \mathbf{H}^{(\ell)} \boldsymbol{\Theta}^{(\ell)} \right)$$

5735 Where  $\boldsymbol{\Theta} \in \mathbb{R}^{d \times d}$  is the parameter matrix. Using  $\mathbf{H}^{(0)} = \mathbf{X}$ , we can  
 5736 use a GCN with  $L$  layers to combine the embeddings in the  $L$ -localized  
 5737 neighborhood of each vertex into a contextualized representation.

5738 Kipf and Welling (2017) evaluate their model on the same NELL dataset  
 5739 used by planetoid with the same 0.1% labeling rate. They train their model  
 5740 by maximizing the log-likelihood of labeled examples. They obtain an  
 5741 accuracy of 66.0%, which is an increase of 4.9 points over planetoid.  
 5742

### 5744 4.3.3 Spatial GCN

5745 Spatial methods directly draw from the comparison with CNN in the spatial  
 5746 domain. As shown by Figure 4.4, the lattice on which a 2-dimensional<sup>77</sup>  
 5747 CNN is applied can be seen as a graph with a highly regular connectiv-  
 5748 ity pattern. In this section, we introduce spatial GCN by following the  
 5749 GraphSAGE model (Hamilton et al. 2017).  
 5750

5751 When computing the activation of a specific node with a CNN, the  
 5752 filter is centered on this node, and each neighbor is multiplied with a  
 5753 corresponding filter element. The products are then aggregated by sum-  
 5754 mation. Spatial GCNs purpose to mimic this process. The main obstacle to  
 5755 generalizing this spatial view of convolutions to graphs is the irregularity  
 5756 of neighborhoods.<sup>78</sup> In a graph, nodes have different numbers of neighbors;  
 5757 a fixed-size filter cannot be used. GraphSAGE proposes several aggregators  
 5758 to replace this product-sum process:

5759 **Mean aggregator** The neighbors are averaged and then multiplied by a  
 5760 single filter  $\mathbf{W}^{(\ell)}$ :

$$5761 \text{aggregate}_{\text{mean}}^{(\ell+1)}(v) = \sigma \left( \mathbf{W}^{(\ell)} \frac{1}{\text{deg}(v) + 1} \sum_{u \in N(v) \cup \{v\}} \mathbf{h}_u^{(\ell)} \right).$$

5762 A spatial GCN using this aggregator is close to the GCN of Kipf and  
 5763 Welling (2017) with  $K = 1$  presented in Section 4.3.2.

5764 **LSTM aggregator** An LSTM (Section 1.3.2.1) is run through all neighbors  
 5765 with the final hidden state used as the output of the layer.  
 5766

$$5767 \text{aggregate}_{\text{LSTM}}^{(\ell+1)}(v) = \text{LSTM}^{(\ell)} \left( \left( \mathbf{h}_u^{(\ell)} \right)_{u \in N(v)} \right)_{\text{deg}(v)}.$$

5772 Since LSTMs are not permutation-invariant, the order in which the  
 5773 neighbors are presented is important.  
 5774

5775 **Pooling aggregator** A linear layer is applied to all neighbors which are  
 5776 then pooled through a max operation.

$$5777 \text{aggregate}_{\text{max}}^{(\ell+1)}(v) = \max \left( \left\{ \mathbf{W}^{(\ell)} \mathbf{h}_u^{(\ell)} + \mathbf{b}^{(\ell)} \mid u \in N(v) \right\} \right).$$

5778

<sup>76</sup> Derivation of the dependency on  $\mathbf{L}_{\text{SYM}}^k$  for the proof of  $K$ -locality:

$$\begin{aligned} \mathbf{x} * \mathbf{w}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}) &= \mathbf{U} \left( \sum_{k=0}^K \theta_k T_k(\boldsymbol{\Lambda}) \right) \mathbf{U}^T \mathbf{x} \\ &= \left( \sum_{k=0}^K \theta_k \mathbf{U} T_k(\boldsymbol{\Lambda}) \mathbf{U}^T \right) \mathbf{x} \\ &= \left( \sum_{k=0}^K \theta_k T_k(\mathbf{L}_{\text{SYM}}) \right) \mathbf{x} \end{aligned}$$

For the last equality, notice that  $\mathbf{L}_{\text{SYM}}^k = (\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T)^k = \mathbf{U} \boldsymbol{\Lambda}^k \mathbf{U}^T$  since  $\mathbf{U}$  is orthogonal. This can also be applied to the (diagonal) constant term.

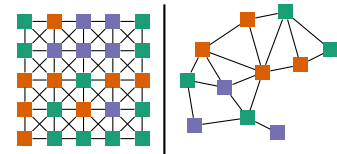


Figure 4.4: Parallel between two-dimensional CNN data and GCN data.

Hamilton et al., “Inductive Representation Learning on Large Graphs” NeurIPS 2017

<sup>77</sup> Even though the same comparison could be made with 1-dimensional CNN as introduced in Section 1.3.1, the similarity is less visually striking. Especially when considering a filter of width 3, in which case the equivalent graph is a simple path graph:  $\cdots \square \square \square \square \cdots$ .

<sup>78</sup> Interestingly enough, this is also a problem with standard CNNs when dealing with values at the edges of the matrix.

Note that the maximum is applied feature-wise.

Using one of these aggregator, a GraphSAGE layer performs the three following operations for all vertices  $v \in V$ :

$$\begin{aligned} \mathbf{a}_v^{(\ell+1)} &\leftarrow \text{aggregate}^{(\ell+1)}(v) \\ \mathbf{h}_v^{(\ell+1)} &\leftarrow \sigma \left( \mathbf{W}_1^{(\ell)} \mathbf{h}_v^{(\ell)} + \mathbf{W}_2^{(\ell)} \mathbf{a}_v^{(\ell+1)} \right) \\ \mathbf{h}_v^{(\ell+1)} &\leftarrow \mathbf{h}_v^{(\ell+1)} / \|\mathbf{h}_v^{(\ell+1)}\|_2. \end{aligned}$$

However, this approach still performs poorly when the graph is irregular.<sup>79</sup> In particular, high-degree vertices—such as “United States” in T-REX as described in Section 4.2—incur significant memory usage. To solve this, GraphSAGE proposes to sample a fixed-size neighborhood for each vertex during training. Their representation is therefore computed from a small number of neighbors. Since  $L$  layers of GraphSAGE produce  $L$ -localized representations, vertices need to be sampled at most at distance  $L$  of the vertex for which we want to generate a representation. Hamilton et al. (2017) propose an unsupervised negative sampling loss to train their GCN such that adjacent vertices have similar representations:

$$\mathcal{L}_{\text{GS}} = \sum_{(u,v) \in E} \log \sigma(\mathbf{z}_v^\top \mathbf{z}_u) - \gamma \mathbb{E}_{v' \sim \mathcal{U}(V)} [\log \sigma(-\mathbf{z}_v^\top \mathbf{z}_{u'})] \quad (4.6)$$

where  $\mathbf{Z} = \mathbf{H}^{(L)}$  is the activation of the last layer and  $\gamma$  is the number of negative samples.

One of the advantages of GraphSAGE compared to the approach of Kipf and Welling (2017) is that it is inductive, whereas the spectral GCN presented in Section 4.3.2 is transductive. Indeed, in the spectral approach, the filter is trained for a specific eigenvectors matrix  $\mathbf{U}$  which depends on the graph. If the graph changes, everything must be re-trained from scratch. In comparison, the parameters learned by GraphSAGE can be reused for a different graph without any problem.

A limitation of GraphSAGE is that the contribution of each neighbor to the representation of a vertex  $v$  is either fixed at  $1 / (\text{deg}(v) + 1)$  (with the mean aggregator) or not modeled explicitly. The same can be observed with the model of Kipf and Welling (2017), where the representation of each neighbor  $u$  is nonparametrically weighted by  $1 / \sqrt{\text{deg}(v) + \text{deg}(u)}$ .

In contrast, graph attention network (GAT, Veličković et al. 2018) proposes to parametrize this weight with a model similar to the attention mechanism presented in Section 1.3.3. The output is built using an attention-like<sup>80</sup> convex combination of transformed neighbors’ representations:

$$\mathbf{h}_v^{(\ell+1)} \leftarrow \sigma \left( \sum_{u \in N(v) \cup \{v\}} \alpha_{vu}^{(\ell)} \mathbf{W}^{(\ell)} \mathbf{h}_u^{(\ell)} \right),$$

where  $\alpha_{vu}^{(\ell)}$ , the attention given by  $v$  to neighbor  $u$  at layer  $\ell$ , is computed using a softmax:

$$\alpha_{vu}^{(\ell)} \propto \exp \text{LeakyReLU} \left( \mathbf{g}^{(\ell)\top} \begin{bmatrix} \mathbf{W}_{\text{GAT}}^{(\ell)} \mathbf{h}_v^{(\ell)} \\ \mathbf{W}_{\text{GAT}}^{(\ell)} \mathbf{h}_u^{(\ell)} \end{bmatrix} \right).$$

As usual, the matrices  $\mathbf{W}$  are parameters, as well as the vector  $\mathbf{g}$  which is used to combine the representations of the two vertices into a scalar weight.

As usual the matrices  $\mathbf{W}_i^{(\ell)}$  are trainable model parameters.

<sup>79</sup> In graph theory, a  $k$ -regular graph is a graph where all vertices have degree  $k$ . By irregular, we mean that the distribution of vertices degrees has high variance; we don’t use the term in its formal “highly irregular” meaning. This is indeed the case in scale-free graphs, as their variance is infinite when  $\gamma < 3$ .

Veličković et al., “Graph Attention Networks” ICLR 2018

<sup>80</sup> Veličković et al. (2018) actually propose to use multi-head attention (Section 1.3.4.1). We describe their model with a single attention head for ease of notation.

LeakyReLU (Maas et al. 2013) is a variant of ReLU where the negative domain is linear with a small slope instead of being mapped to zero:

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0, \\ 0.01x & \text{otherwise.} \end{cases}$$

5833 While GAT and GraphSAGE can be trained in an unsupervised fashion  
 5834 following Equation 4.6, they can also be used as building blocks for larger  
 5835 models, similarly to how we use CNN in Chapter 3. Coupled with the fact  
 5836 that they have a simpler theoretical background and are easier to imple-  
 5837 ment, spatial methods have become ubiquitous to graph-based approaches  
 5838 in the last few years.

5839

5840

#### 5841 4.3.4 GCN on Relation Graphs

5842 All the work introduced in the above sections is about simple undirected  
 5843 graphs  $G = (V, E)$ . In contrast, in Section 4.1, we encoded the relation ex-  
 5844 traction problem on attributed multigraphs  $G = (\mathcal{E}, \mathcal{A}, \varepsilon, \rho)$ . Some works  
 5845 propose to extend GCN to the case of multigraphs, especially when deal-  
 5846 ing with knowledge bases.<sup>81</sup> This is the case of R-GCN (Schlichtkrull et al.  
 5847 2018), a graph convolutional network for relational data. The input graph  
 5848 is not labeled with sentences ( $s$ ) since R-GCN intends to model a knowledge  
 5849 base  $\mathcal{D}_{\text{KB}}$ . This means that while  $G$  is a multigraph, the subgraphs  $G_{(r)}$   
 5850 are simple graphs for all relations  $r \in \mathcal{R}$ . R-GCNs exploit this by using a  
 5851 separate GCN filter for each relation. An R-GCN layer can be defined as:

5852

5853

5854

5855

5856

$$5857 \mathbf{h}_v^{(\ell+1)} \leftarrow \sigma \left( \mathbf{W}_0^{(\ell)} \mathbf{h}_v^{(\ell)} + \sum_{r \in \mathcal{R}} \sum_{u \in N_{\vec{r}}(v)} \mathbf{W}_r^{(\ell)} \mathbf{h}_u^{(\ell)} \right), \quad (4.7)$$

5857 where  $\mathbf{W}_0 \in \mathbb{R}^{d' \times d}$  is used for the (implicit) self-loop, while  $|\mathcal{R}|$  different  
 5858 filters  $\mathbf{W}_r \in \mathbb{R}^{d' \times d}$  are used for capturing the arcs. With highly multi-  
 5859 relational data, the number of parameters grow rapidly since a full matrix  
 5860 needs to be estimated for all relations, even rare ones. To address this issue,  
 5861 Schlichtkrull et al. (2018) propose to either constrain the matrices  $\mathbf{W}_r$  to  
 5862 be block-diagonal, or to decompose them on a small basis  $\mathbf{Z}^{(\ell)} \in \mathbb{R}^{B \times d' \times d}$ :

5863

5864

5865

5866

$$5867 \mathbf{W}_r^{(\ell)} = \sum_{b=1}^B a_{rb}^{(\ell)} \mathbf{Z}_b^{(\ell)},$$

5867 where  $B$  is the size of the basis and  $\mathbf{a}_r$  are the parametric weights for the  
 5868 matrices  $\mathbf{W}_r$ .

5869 Schlichtkrull et al. (2018) evaluate their model on two tasks. First,  
 5870 they evaluate on an entity classification task using a simple softmax layer  
 5871 with a cross-entropy loss on top of the vertex representation at the last  
 5872 layer ( $\mathbf{H}^{(L)}$ ) as defined by Equation 4.7). Second, more closely related to  
 5873 relation extraction, they evaluate on a relation prediction task. Given a  
 5874 pair of entity  $(e_1, e_2) \in \mathcal{E}^2$ , the model must predict the relation  $r \in \mathcal{R}$   
 5875 between them, such that  $(e_1, r, e_2) \in \mathcal{D}_{\text{KB}}$ . To this end, Schlichtkrull et al.  
 5876 (2018) employ the DistMult model which can be seen as a RESCAL model  
 5877 (Section 1.4.2.2) where the interaction matrices are diagonal. The energy  
 5878 of a fact is defined as:

5879

5880

5881

5882

5883

5884

5885

5886

$$5887 \psi_{\text{DistMult}}(e_1, r, e_2) = \mathbf{u}_{e_1}^T \mathbf{C}_r \mathbf{u}_{e_2},$$

5882 where  $\mathbf{u}_e$  is the embedding of the entity at the last layer of the R-GCN:  
 5883  $\mathbf{u}_e = \mathbf{h}_e^{(L)}$  and  $\mathbf{C}_r \in \text{diag}(\mathbb{R}^d)$  is a diagonal matrix parameter. The proba-  
 5884 bility associated to a fact by DistMult is proportional to the exponential  
 5885 of the energy function  $\psi_{\text{DistMult}}$ . Therefore, a missing relation between  
 5886  $e_1, e_2 \in \mathcal{E}$  can be predicted by taking the softmax over relations  $r \in \mathcal{R}$

<sup>81</sup> In this case, the multigraph is simply labeled since the set of relations is finite. In contrast, in the relation extraction problem, the multigraph is attributed. The arcs are associated with a sentence from an infinite set of possible sentences.

Schlichtkrull et al., “Modeling Relational Data with Graph Convolutional Networks” 2018

Note that only the outgoing neighbors  $N_{\vec{r}}$  are taken since for each incoming neighbor labeled  $r$ , there is an outgoing one labeled  $\tilde{r}$ .

Paralleling the notations used for CNNs in Section 1.3.1, we use  $d$  to denote the dimension of embeddings at layer  $\ell$  and  $d'$  for the dimension at layer  $\ell+1$ . More often than not, the same dimension is used at all layers  $d' = d$ . In the following, we use  $d$  as a generic notation for embedding and latent dimensions.

This is similar to the evaluation of TransE reported in Section 1.4.2.3; except that instead of predicting a missing entity in a tuple  $(e_1, r, e_2) \in \mathcal{D}_{\text{KB}}$ , the model must predict the missing relation, assuming  $\mathcal{R}_{1\text{-ADJACENCY}}$  in the process.

5887 of  $\psi_{\text{DistMult}}(e_1, r, e_2)$ . R-GCNs are trained using negative sampling (Sec-  
 5888 tion 1.2.1.3) on the entity classification and relation prediction tasks. This  
 5889 is similar to the training of TransE, where the main difference is that  
 5890 the entity embeddings are computed using R-GCN layers instead of being  
 5891 directly fetched from an entity embedding matrix.

5892 A limitation of R-GCNs is that they only rely on vertices’ represen-  
 5893 tation. Even when the evaluation involves the classification of arcs (as  
 5894 is the case with relation prediction), this is only done by combining the  
 5895 representations of the endpoints (using DistMult).

5896 Several works build upon R-GCN. GP-GNN (H. Zhu et al. 2019) applies  
 5897 a similar model to the supervised relation extraction task. In this case,  
 5898 the graph is attributed with sentences instead of relations; therefore, the  
 5899 weight matrices  $\mathbf{W}_r$  are generated from the sentences instead of using  
 5900 an index of all possible relations. They apply their model to Wikipedia  
 5901 distantly supervised by Wikidata. However, the classification is still made  
 5902 from the representation of the endpoints of arcs. Related work also appears  
 5903 in the *heterogeneous graph* community (Z. Hu et al. 2020; X. Wang et al.  
 5904 2019). Heterogeneous graphs are graphs with labels on both vertices and  
 5905 arcs. The model proposed by Z. Hu et al. (2020) is similar to R-GCN  
 5906 with an attention mechanism more akin to the transformer’s attention  
 5907 (Section 1.3.4.1) than classical attention (Section 1.3.3). The canonical  
 5908 evaluation datasets of this community are citation graphs. Vertices are  
 5909 assigned labels such as “people,” “article” and “conference,” while arcs are  
 5910 labeled with a small number of domain-specific relations: *author*, *published*  
 5911 *at*, *cite*, etc. The evaluation task typically corresponds to entity prediction.

5912

5913

#### 5914 4.3.5 Weisfeiler–Leman Isomorphism Test

5915

5916 In this section, we introduce the theoretical background of GCNs. This  
 5917 is of particular interest to us since this theoretical background is more  
 5918 closely related to unsupervised relation extraction than GCNs can be at first  
 5919 glance. As stated in the introduction to the thesis, relations emerge from  
 5920 repetitions. In particular, we expect that two identical (sub-)graphs convey  
 5921 the same relations. However, testing whether two graphs are identical is a  
 5922 complex problem. Indeed, we have to match each of the  $n$  vertices of the  
 5923 first graph to one of the  $n$  possibilities in the second graph. Naively, we  
 5924 need to try all  $n!$  possibilities. This is known as the graph isomorphism  
 5925 problem. Two simple graphs  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$  are said to be  
 5926 isomorphic ( $G_1 \simeq G_2$ ) iff there exists a bijection  $f: V_1 \rightarrow V_2$  such that  
 5927  $(u, v) \in E_1 \iff (f(u), f(v)) \in E_2$ . Figure 4.5 gives an example of two  
 5928 isomorphic graphs.

5929 The various GCN methods introduced thus far can be seen as generaliza-  
 5930 tions of the Weisfeiler–Leman<sup>82</sup> isomorphism test (Weisfeiler and Leman  
 5931 1968), which tests whether two graphs are isomorphic. The  $k$ -dimensional  
 5932 Weisfeiler–Leman isomorphism test ( $k$ -dim WL) is a polynomial-time al-  
 5933 gorithm assigning a color to each  $k$ -tuple of vertices<sup>83</sup> such that two iso-  
 5934 morphic graphs have the same coloring. With a bit of work, the general  
 5935  $k$ -dim WL algorithm can be implemented in  $O(k^2 n^{k+1} \log n)$  (Immerman  
 5936 and Lander 1990). However, there exist pairs of graphs that are not iso-  
 5937 morphic, yet are assigned with the same coloring by the Weisfeiler–Leman  
 5938 test (Cai et al. 1992). At the time of writing, the precise membership of  
 5939 the graph isomorphism problem with respect to the polynomial complex-  
 5940 ity classes is still conjectural. No polynomial-time algorithm nor reduction

H. Zhu et al., “Graph Neural Networks with Generated Parameters for Relation Extraction” ACL 2019

Z. Hu et al., “Heterogeneous Graph Transformer” www 2020

X. Wang et al., “Heterogeneous Graph Attention Network” www 2019

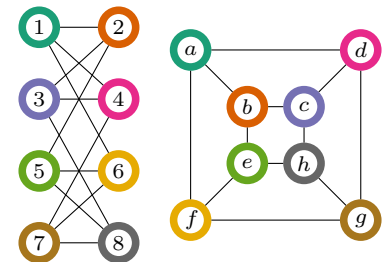


Figure 4.5: Example of isomorphic graphs. Each vertex  $i$  in the first graph corresponds to the  $i$ -th letter of the alphabet in the second graph. Alternatively, these graphs have nontrivial automorphism, for example, by mapping vertex  $i$  to vertex  $9 - i$ .

<sup>82</sup> Often spelled Weisfeiler–Lehman, Babai (2016) indicates that Andrei Leman preferred to transliterate his name without an “h.”

Weisfeiler and Leman, “The reduction of a graph to canonical form and the algebra which appears therein” NTI 1968

<sup>83</sup> An ordered sequence of  $k$  vertices, that is an element of  $V^k$ , not necessarily connected.

Cai et al., “An optimal lower bound on the number of variables for graph identification” Combinatorica 1992



**algorithm** WEISFEILER–LEMAN

Inputs:  $G = (V, E)$  graph  
 $k$  dimensionality  
Output:  $\chi_\infty$  coloring of  $k$ -tuples

▷ Initialization ◁

$\ell \leftarrow 0$

**for all**  $\mathbf{x} \in V^k$  **do**

$\chi_0(\mathbf{x}) \leftarrow \text{iso}(\mathbf{x})$

▷ Main Loop ◁

**repeat**

$\ell \leftarrow \ell + 1$

$\mathcal{J}_\ell \leftarrow$  new color index

**for all**  $\mathbf{x} \in V^k$  **do**

$c_\ell(\mathbf{x}) \leftarrow \{ \chi_{\ell-1}(\mathbf{y}) \mid \mathbf{y} \in N^k(\mathbf{x}) \}$

$\chi_\ell(\mathbf{x}) \leftarrow$  index of  $(\chi_{\ell-1}(\mathbf{x}), c_\ell(\mathbf{x}))$  in  $\mathcal{J}_\ell$

**until**  $\chi_\ell = \chi_{\ell-1}$

**output**  $\chi_\ell$

Algorithm 4.2: The Weisfeiler–Leman isomorphism test. The double braces  $\{ \}$  denote a multiset. Since  $\mathcal{J}_\ell$  is indexed with the previous coloring  $\chi_{\ell-1}(\mathbf{x})$  of the vertices—alongside  $c_\ell(\mathbf{x})$ —the number of color classes is strictly increasing until the last iteration when it remains constant. Since the last coloring is stable, we refer to it as  $\chi_\infty$ .

from NP-complete problems are known. This makes graph isomorphism one of the prime candidates for the NP-intermediate complexity class.<sup>84</sup>

The general  $k$ -dim WL test is detailed in Algorithm 4.2. It is a refinement algorithm, which means that at a given iteration, color classes can be split, but two  $k$ -tuples with different colors at iteration  $\ell$  can't have the same color at iteration  $\ell' > \ell$ . Initially, all  $k$ -tuples  $x$  are assigned a color according to their isomorphism class  $\text{iso}(x)$ . We define the isomorphism class through an equivalence relation. For two  $k$ -tuples  $\mathbf{x}, \mathbf{y} \in V^k$ ,  $\text{iso}(x) = \text{iso}(y)$  iff:<sup>85</sup>

- $\forall i, j \in [1, \dots, k] : x_i = x_j \iff y_i = y_j$
- $\forall i, j \in [1, \dots, k] : (x_i, x_j) \in E \iff (y_i, y_j) \in E$

Intuitively, this checks whether  $x_i \mapsto y_i$  is an isomorphism for the subgraphs built from the  $k$  vertices  $\mathbf{x}$  and  $\mathbf{y}$ . This is not the same as the graph isomorphism problem since here, the candidate isomorphism is given, we don't have to test the  $k!$  possibilities.

The coloring of  $\mathbf{x} \in V^k$  is refined at each step by juxtaposing it with the coloring of its neighbors  $N^k(\mathbf{x})$ . We need to reindex the new colors at each step since the length of the color strings would grow exponentially otherwise. The set of neighbors<sup>86</sup> of a  $k$ -tuple for  $k \geq 2$  is defined as:

$$N^k(\mathbf{x}) = \{ \mathbf{y} \in V^k \mid \exists i \in [1, \dots, k] : \forall j \in [1, \dots, k] : j \neq i \implies x_j = y_j \}.$$

In other words, the  $k$ -tuples  $\mathbf{y}$  neighboring  $\mathbf{x}$  are those differing by at most one vertex with  $\mathbf{x}$ .

The 1-dim WL test is also called the *color refinement* algorithm. In this case,  $N^1(x)$  is simply  $N(x)$  the set of neighbors of  $x$ . The isomorphism class of a single vertex is always the same, so  $\chi_0$  assigns the same color to all vertices. The first iteration of the algorithm groups vertices according to their degree (the multiplicity of the sole element in the multiset  $c_1(x)$ ). The second iteration  $\chi_2$  then colors each vertex according to its degree  $\chi_1$  and the degree of its neighbors  $c_2$ . And so on and so forth until  $\chi$  does not change anymore.

<sup>84</sup> The class of NP problems neither in P nor NP-complete. It is guaranteed to be non-empty if  $P \neq NP$ . Clues for the NP-intermediateness of the graph isomorphism problem can be found in the fact that the counting problem is in NP (Mathon 1979) and more recently, from the fact that a quasi-polynomial algorithm exists (Babai 2015).

<sup>85</sup> To avoid having to align two colorings, the WL algorithm is usually run on the disjoint union of the two graphs. So, strictly speaking, it tests for automorphism (isomorphic endomorphism). Therefore we can assume  $\mathbf{x}$  and  $\mathbf{y}$  are from the same vertex set  $V$ .

<sup>86</sup> Note that the kind of neighborhood defined by  $N^k$  completely disregards the edges in the graph. For this reason, it is sometimes called the *global neighborhood*.

5995 The GCN introduced in the previous sections can be seen as variants  
 5996 of the 1-dim WL algorithm where the index  $\mathcal{J}_\ell$  is replaced with a neu-  
 5997 ral network such as  $\text{aggregate}_{\text{mean}}^{(\ell)}$  given in Section 4.3.3. In this case  $\chi_\ell$   
 5998 corresponds to  $\mathbf{H}^{(\ell)}$  the activations at layer  $\ell$ .  
 5999

6000

## 6001 4.4 Proposed Approaches

6002

6003 We now turn to the graph-based models we propose to leverage infor-  
 6004 mation from the structure of the dataset. Let us quickly summarize the  
 6005 context in which we inscribe our work. We have access to two kinds of  
 6006 features: linguistic—from the sentence—and topological—from the graph.  
 6007 Unsupervised relation extraction methods do not fully exploit graph neigh-  
 6008 borhoods.<sup>87</sup> Supervised methods such as EPGNN and GP-GNN do, even  
 6009 though the information present in the graph is more important in the un-  
 6010 supervised setting. Indeed, the relational information is mostly extractable  
 6011 from the sentences and entities alone. While extra information from topo-  
 6012 logical features can still be used by supervised models, it is not essential.  
 6013 On the other hand, in the unsupervised setting, the main issue is to iden-  
 6014 tify the relational information in the sentence, to distinguish it from other  
 6015 semantic contents. As we show in Section 4.2, this relational information  
 6016 is also present in the topological features (the neighborhood of a sample).  
 6017 This can be useful in two ways:

6018

- 6019 1. Use both pieces of information jointly, linguistic and topological:  
 6020 “the more features, the better.” This is what supervised models do.  
 6021
- 6022 2. Use the topological features to identify the relational information in  
 6023 the linguistic features.

6024

6025 In Section 4.4.1, we exploit the first point by adding a GCN to the  
 6026 matching the blanks model (MTB, Section 2.5.6). In Section 4.4.2, we show  
 6027 that topological features can be used without training a GCN. This also  
 6028 serves as an introduction to Section 4.4.3, which proposes an unsupervised  
 6029 loss following the second point above; it exploits the fact that relation  
 6030 information is present in both linguistic and topological features.

6031

### 6032 4.4.1 Using Topological Features

6033

6034 In this section, we seek to use topological information as additional fea-  
 6035 tures for an existing unsupervised model: matching the blanks (MTB). The  
 6036 usefulness of these features lies in the fact that many relations are “typed”:  
 6037 e.g. they only accept geographical locations as objects and only people as  
 6038 subjects (such as *born in*). This can be captured by looking at the neigh-  
 6039 borhood of each entity, which can be seen as a “soft” version of  $\mathcal{R}_{\text{TYPE}}$   
 6040 (“relations are typed,” Section 2.5.3).

6041

6042 A straightforward approach is to parallel the construction of R-GCN  
 6043 (Section 4.3.4): use a GCN-like encoder followed by a relation classifier—in  
 6044 the case of R-GCN, DistMult. In effect, this corresponds to taking MTB  
 6045 and augmenting it with a GCN to process neighboring samples. As a re-  
 6046 minder, MTB uses a similarity-based loss where each unsupervised sample  
 6047  $(s, e) \in \mathcal{D}$  is represented by  $\text{BERTcoder}(s)$ . In this model, the information  
 6048 lies on the arcs. In order to use a GCN model, we transform our graph  
 $G = (\mathcal{E}, \mathcal{A}, \varepsilon, \rho, \varsigma)$  such that the information lies on the vertices instead.

<sup>87</sup> As explained in Section 4.1, MTB does use close neighborhoods as contrast during training, but not for inference.

6049 This transformed graph is called the *line graph* of  $G$  and noted  $L(G)$ . An  
 6050 illustration for simple undirected graphs is provided in Figure 4.6. For a  
 6051 directed (multi)graph, it is defined as follows:

$$L(G) = (\mathcal{A}, \mathfrak{A}, \varepsilon, \varsigma)$$

$$\mathfrak{A} = \{ (a_1, a_2) \in \mathcal{A}^2 \mid \varepsilon_2(a_1) = \varepsilon_1(a_2) \}.$$

6056 In other words, each arc becomes a vertex and an arc  $a_1 \rightarrow a_2$  is present if  
 6057 and only if  $a_1$  and  $a_2$  form a directed path of length 2. The neighborhood of  
 6058 each sample (arc is the original  $G$ ) is still defined as all other samples with  
 6059 at least one entity in common since by construction for all v-structures  
 6060  $e_1 \xrightarrow{a_1} e_2 \xleftarrow{a_2} e_3$ , there exists a directed path  $e_1 \xrightarrow{a_1} e_2 \xrightarrow{a_2} e_3$  in the original graph  
 6061  $G$ . This construction is actually similar to the one of EPGNN introduced in  
 6062 Section 2.4.5. The main difference is that each vertex in  $L(G)$  corresponds  
 6063 to a sample in  $\mathcal{D}$ , while an EPGNN graph groups samples by entity pairs  
 6064 into a single vertex.

6065 The standard loss and training algorithm of MTB as defined by Equation 2.10 can be reused as is, we only need to redefine the similarity function (Equation 2.9):

$$\text{sim}(a, a', G) = \sigma \left( \begin{array}{c} \text{BERTcoder}(\varsigma(a))^\top \text{BERTcoder}(\varsigma(a')) \\ + \lambda \text{GCN}(L(G))_a^\top \text{GCN}(L(G))_{a'} \end{array} \right), \quad (4.8)$$

6073 where  $\lambda$  is a hyperparameter weighting the topological-based prediction  
 6074 over the sentence-based one. At the input of the GCN, the vertices are  
 6075 labeled using the same sentence encoder:  $\mathbf{x}_a = \text{BERTcoder}(\varsigma(a))$ .

6076 The only difference between MTB and the MTB-GCN hybrid we propose  
 6077 is the additional  $\lambda$ -weighted term in Equation 4.8. We use this model to  
 6078 evaluate whether topological features can be exploited by an existing un-  
 6079 supervised relation extraction loss. It tells us how much can be gained from  
 6080 the “adding more features” aspect of graph-based methods and contrast  
 6081 it with the new topology-aware loss design we propose in Section 4.4.3.

#### 6082 6083 4.4.2 Nonparametric Weisfeiler–Leman Iterations

6085 The losses used to train unsupervised GNNs usually make the hypothesis  
 6086 that linked vertices should have similar representations. This can be seen  
 6087 in  $\mathcal{L}_{\text{GS}}$  (Equation 4.6), which seeks to maximize the dot product between  
 6088 the representations of adjacent vertices. While this hypothesis might be  
 6089 helpful for most problems on which GNNs are applied, this is clearly not  
 6090 the case for relation extraction. In Section 4.4.1, we introduced a first  
 6091 simple solution to this problem is to replace the loss used by the GNN  
 6092 with a standard unsupervised relation extraction loss. However, it is also  
 6093 possible to design an unsupervised loss from the theoretical foundation of  
 6094 GCN: the Weisfeiler–Leman isomorphism test. To this end, we propose to  
 6095 build a model relying on the following hypothesis:

6097 **Weak Distributional Hypothesis on Relation Extraction Graph:**  
 6098 *Two arcs conveying similar relations have similar neighborhoods.*

6099 Note that we dubbed this version of the distributional hypothesis *weak*  
 6100 since we only state it in one direction, the converse having several counter-  
 6101 examples. For example, sentences about the place of birth and the place  
 6102

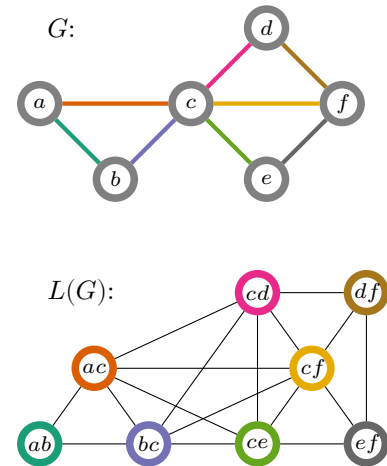


Figure 4.6: Example of line graph construction. Each edge  $x-y$  in the simple undirected graph  $G$  corresponds to the vertex  $xy$  with the same color in the graph  $L(G)$ . Two vertices in  $L(G)$  are connected iff the corresponding edges share an endpoint in  $G$ . In directed graphs, the two arcs further need to be in the same direction in  $G$  for an arc to exist in  $L(G)$ .

of death of a person tend to have similar neighborhoods despite conveying different relations.<sup>88</sup> To distinguish these kinds of relations with similar neighborhoods, we have to rely on sentence representations.<sup>89</sup>

Following this hypothesis, we first propose a simple parameter-less approach based on the Weisfeiler–Leman isomorphism test (Section 4.3.5). We can say that two neighborhoods are similar if they are isomorphic. Therefore, we can enforce the hypothesis above by ensuring that if two neighborhoods are assigned similar coloring by the WL algorithm, they convey similar relations. In the relation extraction problem, contrary to much of the related work presented in Section 4.3, we have data on the arcs of the graph, not on the vertices. This means that instead of using the 1-dimensional Weisfeiler–Leman algorithm, we use the 2-dimensional version. In other words, instead of coloring the vertices, we color the arcs since our problem is to label them with a relation.

The initial coloring  $\chi_0(a)$  is initialized as the isomorphism class of a sample  $a \in \mathcal{A}$ . We can define this isomorphism class using  $\text{BERTcoder}(a)$ , which means that the initial representation of a sample will simply be the sentential representation of the sample. The difficult task is to define the re-indexing of colors as performed by  $\mathcal{J}$  in Algorithm 4.2. This is difficult since the original WL algorithm is defined on a discrete set of colors, while we need to manipulate distributed representations of sentences.

If we want to produce clear-cut relation classes, we can use a hashing algorithm on sentence representations such as the one proposed for graph kernels by Morris et al. (2016). However, we focus on a few-shot evaluation in order to compare with MTB and to avoid errors related to knowledge base design as described in Section 2.5.1.2. In this case, we only need to be able to compare the colors of two different samples, measuring how close they are to each other. Let us define  $\mathcal{N} : \mathcal{A} \rightarrow 2^{\mathcal{A}}$  the function mapping an arc to the set of its neighbors. Formally, for  $a \in \mathcal{A}$ ,  $\mathcal{N}(a) = \{a' \in \mathcal{A} \mid \varepsilon(a) \cap \varepsilon(a') \neq \emptyset\}$ . In other words,  $\mathcal{N}$  in  $G$  corresponds to the neighbors function  $N$  in the line graph  $L(G)$ . Since  $\mathcal{A}$  can be seen as the set of samples,  $\mathcal{N}(a)$  can be seen as the set of samples with at least one entity in common with  $a$ . To enforce the weak distributional hypothesis on graphs stated above, we take two first-order neighborhoods  $\mathcal{N}(a), \mathcal{N}(a') \subseteq \mathcal{A}$  and define a distance between them. This corresponds to comparing two empirical distributions of sentence representations<sup>90</sup> that have an entity in common with  $a$  and  $a'$ . This can be done using the 1-Wasserstein distance between the two neighborhoods since they can be seen as two distributions of Dirac deltas in  $\text{BERTcoder}$  representation space.<sup>91</sup> This needs to be done for the two entities, which correspond to the in-arc-neighbors  $\mathcal{N}_{\leftarrow}$  and out-arc-neighbors  $\mathcal{N}_{\rightarrow}$ . While this is 1-localized, we can generalize this encoding to be  $K$ -localized by defining the  $k$ -sphere centered on an arc  $a$ , where the 1-sphere corresponds to  $\mathcal{N}$ :

$$\begin{aligned} \mathcal{S}_{\rightarrow}(a, 0) &= \{a\} \\ \mathcal{S}_{\rightarrow}(a, k) &= \{x \in \mathcal{A} \mid \exists y \in \mathcal{S}_{\rightarrow}(a, k-1) : \varepsilon_1(x) = \varepsilon_2(y)\}. \end{aligned}$$

This sphere can be embedded using  $\text{BERTcoder}$ , which corresponds to retrieving its initial coloring:

$$\mathfrak{S}_{\rightarrow}(a, k) = \{\text{BERTcoder}(s(x)) \in \mathbb{R}^d \mid x \in \mathcal{S}_{\rightarrow}(a, k)\}.$$

We can thereafter define the  $K$ -localized out-neighborhood of  $a \in \mathcal{A}$  as the sequence of  $\mathfrak{S}_{\rightarrow}(a, k)$  for all  $k = 1, \dots, K$ . The in-neighborhood is defined

<sup>88</sup> The neighborhoods are somewhat dissimilar in that “notable” people tend to die in places with more population than their birthplace. However, whether current models can pick this up from other kinds of regularity in a dataset is dubious.

<sup>89</sup> This can partly explain the conditional entropy  $H(r_2 \mid r_1, r_3) \approx 1.06$  bits given in Section 4.2.

The astute reader might have noticed that the 2-dimensional WL isomorphism test as described in Algorithm 4.2 loops over pairs of vertices, not arcs. This is impractical in our relation extraction graph, which is particularly sparse—the number of arcs  $m$  is far larger than the number of vertices  $n$ . The extra (unlinked) entity pairs considered by Algorithm 4.2 are usually referred to as *anti-arcs*. Ignoring anti-arcs leads to the local Weisfeiler–Leman isomorphism tests since only the “local neighborhood” is considered. Other intermediate approaches are possible, sometimes referred to as the *glocalized* variants of Weisfeiler–Leman. See Morris et al. (2020) for an example of application to graph embeddings. Alternatively, our proposed approach can be seen as a 1-dimensional Weisfeiler–Leman isomorphism test applied to the line graph.

<sup>90</sup> We are comparing sentence representations and not directly sentences since the initial coloring  $\chi_0$  has been defined using  $\text{BERTcoder}$ .

<sup>91</sup> Wasserstein distance has the advantage of working on distributions with disjoint supports.

6157 similarly. Finally, the distance between two samples  $a, a' \in \mathcal{A}$  can be  
6158 defined as:

$$6159 \quad d(a, a'; \boldsymbol{\lambda}) = \sum_{k=0}^K \frac{\lambda_k}{2} \sum_{o \in \{\leftarrow, \rightarrow\}} W_1(\mathfrak{S}_o(a, k), \mathfrak{S}_o(a', k)), \quad (4.9)$$

6163 where  $W_1$  designates the 1-Wasserstein distance, and  $\boldsymbol{\lambda} \in \mathbb{R}^{K+1}$  weights  
6164 the contribution of each sphere to the final distance value. In particular  
6165  $\lambda_0$  parametrizes how much the linguistic features should weight compared  
6166 to topological features.<sup>92</sup>

6167 To relate this function back to our original re-coloring problem, the  
6168 distance  $d$  up to  $K$  can be seen as a distance on  $\chi_K$ , the coloring assigned at  
6169 step  $K$ . Indeed, if  $d(a, a', \boldsymbol{\lambda}) = 0$  then  $\chi_K(a) = \chi_K(a')$ . However, while two  
6170 colors are either equal or not in the original algorithm, the distance  $d$  gives  
6171 a topology to the set of arcs. We don't directly compute a hard-coloring of  
6172 2-tuples. The closest thing to a coloring  $\chi$  in our algorithm is the sphere  
6173 embedding  $\mathfrak{S}$ , which in fact, is more akin to  $c$  in Algorithm 4.2. In other  
6174 words, we skip the re-indexing step of the Weisfeiler–Leman algorithm to  
6175 deal with the continuous nature of sentence embeddings at the cost of a  
6176 higher computational cost.

6177 Combining a Wasserstein distance with Weisfeiler–Leman was already  
6178 proposed for graph kernels (Togninalli et al. 2019). However, this was  
6179 applied to a simple graph without attributed edges, and it was unrelated to  
6180 any information extraction task. For unsupervised relation extraction, the  
6181 distance function  $d$  can directly be used to compute the similarity between  
6182 query and candidates samples in a few-shot problem (Section 2.5.1.2).  
6183 Since the number of arcs at distance  $k$  grows quickly in a scale-free graph,<sup>93</sup>  
6184 we either need to keep  $K$  low or employ sampling strategies similarly  
6185 to GraphSAGE (Section 4.3.3). Furthermore, the Wasserstein distance is  
6186 hard to compute exactly; entropic regularization of the objective has been  
6187 proposed. In particular,  $W_1$  can be efficiently computed with Sinkhorn  
6188 iterations (Cuturi 2013).

### 6191 4.4.3 Refining Linguistic and Topological Features

6192 While the nonparametric method presented in the previous section man-  
6193 ages to consider both the linguistic and topological features, it processes  
6194 them in isolation. In this section, we propose a scheme that allows both  
6195 the encoder of linguistic and topological features to adapt to each other  
6196 in a training process. Conceptually, this is somewhat similar to SelfORE  
6197 (Section 2.5.7). As a reminder, SelfORE is a clustering method that purifies  
6198 relation clusters by optimizing BERTcoder such that samples with close lin-  
6199 guistic forms are pushed closer. In our scheme, we propose to refine both  
6200 linguistic and topological features with respect to each other. In this way  
6201 we hope to both enforce  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$  and the following assumption:

6203 **Assumption  $\mathcal{H}_{1\text{-NEIGHBORHOOD}}$ :** *Two samples with the same neighborhood*  
6204 *in the relation extraction graph convey the same relation.*

$$6205 \quad \forall a, a' \in \mathcal{A}: \mathcal{N}(a) = \mathcal{N}(a') \implies \rho(a) = \rho(a')$$

6207 Note that this is the converse of the weak distributional hypothesis on  
6208 relation extraction graph stated in Section 4.4.2. We need to make the  
6209 modeling hypothesis in this direction since in the unsupervised relation  
6210 extraction problem, we do not have access to relations and therefore can't

To be precise Equation 4.9 defines a distance between samples from the Euclidean distances between neighboring samples—that is samples with an entity in common. The distance  $W_1$  is the cost of the optimal transport plan between two sets of Dirac deltas corresponding to the neighborhoods of the samples.

<sup>92</sup> The 1-Wasserstein distance is defined on top of a metric space; therefore, the difference between two neighbors must be defined using the Euclidean distance. We can't use dot product as usually done with BERT representations (see for example Equation 2.9). However, we can slightly change Equation 4.9 to use the dot product for the computation of the linguistic similarity (the term  $k = 0$ ). In this case, however,  $d$  would no longer satisfy the properties of a metric.

<sup>93</sup> Remember that the diameter of the (scale-free) graph is in the order of  $\log \log n$ .

As a reminder,  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$  states that two samples with similar contextualized embeddings convey similar relations. See Appendix B.

6211 enforce an hypothesis between samples conveying the same relations. We  
 6212 posit that by balancing  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$  and  $\mathcal{H}_{1\text{-NEIGHBORHOOD}}$  we are able to  
 6213 exploit the structure induced by both sources information in an unsuper-  
 6214 vised samples  $(s, e) \in \mathcal{D}$ : the sentence  $s$  and entities  $e$ , whereas SelfORE  
 6215 only relies on the sentence  $s$ .

6216 To define the topological and linguistic distance between two sam-  
 6217 ples, we use the distance function defined by Equation 4.9. For computa-  
 6218 tional reasons, we set  $K = 1$ , which means that our model is 1-localized.  
 6219 The linguistic distance is simply the distance between the BERTcoder of  
 6220 the samples’ sentences. In other words, it is  $d(a, a'; [1, 0]^T)$ . On the other  
 6221 hand, the topological distance can be defined as the distance between the  
 6222 two neighborhoods, in other words,  $d(a, a'; [0, 1]^T)$ . We propose to train  
 6223 BERTcoder such that these two distances coincide more. In practice, this  
 6224 can be achieved with a triplet loss similar to the one used by TransE  
 6225 (Section 1.4.2.3). Given three arcs  $\mathbf{a} \in \mathcal{A}^3$ , we ensure the two distances  
 6226 are similar between the two first arcs  $a_1$  and  $a_2$ , and we contrast these  
 6227 distances using the third arc  $a_3$ . This translates to the following loss:

$$6228 \mathcal{L}_{\text{LT}}(a_1, a_2, a_3) = \max \left( \begin{array}{l} 0, \zeta + 2(d(a_1, a_2, [1, 0]^T) - d(a_1, a_2, [0, 1]^T))^2 \\ 6229 \quad - (d(a_1, a_2, [1, 0]^T) - d(a_1, a_3, [0, 1]^T))^2 \\ 6230 \quad - (d(a_1, a_3, [1, 0]^T) - d(a_1, a_2, [0, 1]^T))^2 \end{array} \right),$$

6231 where  $\zeta > 0$  is a hyperparameter defining the maximum margin we seek to  
 6232 enforce between the true distance-error and the negative distance-error. By  
 6233 randomly sampling arcs triplets  $\mathbf{a} \in \mathcal{A}^3$ , we can fine-tune a BERTcoder in  
 6234 an unsupervised fashion such that it captures both linguistics and topolog-  
 6235 ical features. During evaluation, the procedure described in Section 4.4.2  
 6236 can be reused, such that both linguistic representations refined by the  
 6237 topological structure and the topological representations refined by the  
 6238 linguistic structure are used jointly. However, both distances could be used  
 6239 independently, for example if a sample contains unseen entities, or on the  
 6240 contrary if we want to assess which relation links two entities without any  
 6241 supporting sentence.

6242

6243

6244

6245

6246

6247

6248

## 6249 4.5 Experiments

6249 Matching the blanks was trained on a huge unsupervised dataset that is  
 6250 not publicly available (Soares et al. 2019). To ensure reproducibility, we  
 6251 instead attempt to train on T-REX (Section C.7, Elsahar et al. 2018). The  
 6252 evaluation is done in the few-shot setting (Section 2.5.1.2) on the FewRel  
 6253 dataset (Section C.2) in the 5-way 1-shot setup. Our code is available at  
 6254 <https://esimon.eu/repos/gbure>.

6255 The BERTcoder model we use is the entity markers–entity start de-  
 6256 scribed in Section 2.3.7, based on a `bert-base-cased` transformer. We use  
 6257 a BERTcoder with no post-processing layer for the standalone BERT model.  
 6258 The MTB model is followed by a layer norm even during pre-training as de-  
 6259 scribed by Soares et al. (2019). The MTB similarity function remains a dot  
 6260 product but was rescaled to be normally distributed. When augmenting  
 6261 MTB with a GCN, we tried both the Chebyshev approximation described in  
 6262 Section 4.3.2 and the mean aggregator of Section 4.3.3, however we were  
 6263 only able to train de Chebyshev variant at the time of writing. The non-  
 6264 parametric WL algorithm uses a dot product for linguistic similarity and

Intuitively, we want to optimize the mean squared error (MSE) between the linguistic and topological features of all pairs of arcs  $(d(a_1, a_2, [1, 0]^T) - d(a_1, a_2, [0, 1]^T))^2$ . However, this loss could be optimized by encoding all arcs into a single point. The output of BERTcoder would then be constant. Therefore, we need to regularize the MSE loss such that distances that shouldn’t be close are not. This is the point of the triplet loss; we contrast the positive distance delta with a negative one. While  $d(a_1, a_2, [1, 0]^T)$  and  $d(a_1, a_2, [0, 1]^T)$  should be close to each other (because of  $\mathcal{H}_{1\text{-NEIGHBORHOOD}}$ ), they shouldn’t be close to any distance involving a third sample  $a_3$ . This ensures that our model does not collapse.

Elsahar et al., “T-REX: A Large Scale Alignment of Natural Language with Knowledge Base Triples” LREC 2018

a Euclidean 1-Wasserstein distance for topological distance; the hyperparameters are  $\lambda = [-1, 0.2]^T$ .

We report our results in Table 4.2. The given numbers are accuracies on the subset of FewRel with at least one neighbor in T-REX. The accuracies on the whole dataset are 73.74% for linguistic features alone (BERT) and 77.54% for MTB. Our results for MTB are still slightly below what Soares et al. (2019) report because of the BERT model size mismatch and the smaller pre-training dataset. The result gap is within expectations, as already reported by other works that used a similar setup on the supervised setup (Qu et al. 2020). On the other hand, our accuracy for a standalone BERT is higher than what Soares et al. (2019) report; we suspect this is due to our removal of the randomly initialized post-processing layer.

The top half of Table 4.2 reports results for nonparametric models. These models were not trained for the relation extraction task; they simply exploit an MLM-pretrained BERT in clever ways. As we can see, while topological features are a bit less expressive to extract relations by themselves, they still contain additional information that can be used jointly with linguistic features—this is what the nonparametric WL model does.

For parametric models, we have difficulties training on T-REX because of its relative small size. In practice 66.89% of FewRel entities are already mentioned in T-REX. However, a standard 5-way 1-shot problem contains  $(1 + 5) \times 2 = 12$  different entities. We measure the empirical probability that all entities of a few-shot problem are connected in T-REX to be around 0.54%. Furthermore, we observe that MTB augmented with a GCN performs worse than a standalone MTB despite adding a single linear layer to the parameters (the BERTcoder of the linguistic and topological distances are shared). These are still preliminary results, however, it seems the small size of T-REX coupled with the large amount of additional information presented to the model cause it to overfit on the train data. We observe a similar problem with the triplet loss model of Section 4.4.3. At the time of writing, our current plan is to attempt training on a larger graph, similar to the unsupervised dataset of Soares et al. (2019).

Model	Accuracy
Linguistic (BERT)	69.46
Topological ( $W_1$ )	65.75
Nonparametric WL	72.18
MTB	78.83
MTB GCN-Chebyshev	76.10

Table 4.2: Preliminary results for FewRel valid accuracies of graph-based approaches. To better evaluate the efficiency of topological features, we report results on the subset of the dataset that is connected in T-REX.

## 4.6 Conclusion

In this chapter, we explore aggregate approaches to unsupervised relation extraction using graphs. In Section 4.2, we show that a large amount of information can be leveraged from the neighborhood of a sample. This, together with the observation that previous unsupervised methods always ignored the neighborhood of a sample at inference, opens a new research direction for unsupervised methods. In Section 4.4, we propose several models that make use of the neighborhood information. In particular, we propose a novel unsupervised training loss in Section 4.4.3, which makes very few modeling assumptions while still being able to exploit the neighborhood information both at training and prediction time.

Our contributions lie in using a multigraph with arcs attributed with sentences (Sections 4.1), our method to approximate the quantity of information extractable from this graph (Sections 4.2) and our proposed approach to utilize this additional information (Section 4.4). Despite encouraging early results showing the soundness of using the relation extraction graph, at the present time we only improved nonparametric models. More

6319 experimentation is still needed to fully exploit topological information.

6320

6321

6322

6323

6324

6325

6326

6327

6328

6329

6330

6331

6332

6333

6334

6335

6336

6337

6338

6339

6340

6341

6342

6343

6344

6345

6346

6347

6348

6349

6350

6351

6352

6353

6354

6355

6356

6357

6358

6359

6360

6361

6362

6363

6364

6365

6366

6367

6368

6369

6370

6371

6372



## Conclusion

During this Ph.D. candidacy, I—mostly<sup>94</sup>—focused on the study of unsupervised relation extraction. In this task, given a set of tagged sentences and pairs of entities, we seek the set of conveyed facts  $(e_1, r, e_2)$ , such that  $r$  embodies the relationship between  $e_1$  and  $e_2$  expressed in some sample. To tackle this task, we follow two main axes of research: first, the question of how to train a deep neural network for unsupervised relation extraction; second, the question of how to leverage the structure of an unsupervised dataset to gain additional information for the relation extraction task.

## Summary of Contributions

For more than a decade now, the field of machine learning has been overrun by deep learning approaches. Since I started working on unsupervised relation extraction in late 2017, the task followed the same fate. The VAE model of Marcheggiani and Titov (2016) started introducing deep learning methods to the task. However, it was still limited by a sentence representation based on hand-engineered features. My first axis of research was to partake in this deep learning transition (Chapter 3). Subsequently, the use of deep learning was made simpler with the replacement of CNN and LSTM-based models with pre-trained transformers. Indeed, a model like BERT (Devlin et al. 2019) performs reasonably well on unsupervised relation extraction “out of the box.” This was exploited by others, in the clustering setup by SelfORE (X. Hu et al. 2020), and in the few-shot setup by MTB (Soares et al. 2019). My second axis of research was to exploit the regularities of the dataset to leverage additional information from its structure (Chapter 4). While some works already used this information in supervised relation extraction (Chen et al. 2006; Zhao et al. 2019), unsupervised models made no attempt at modeling it explicitly. Our proposed approaches are based on a graph representation of the dataset. As we have shown, they inscribe themselves in a general revival of graph-based approaches in deep learning (Hamilton et al. 2017; Kipf and Welling 2017). We now describe the three main contributions we can draw from our work.

### Literature review with formalized modeling assumptions.

In Chapter 2, we presented relevant relation extraction models from the late 1990s until today. We first introduced supervised approaches, which we split into two main blocks:

*Sentential methods* extract a relation for each sample in isolation. In this setup, there is no difference between evaluating a model on a single dataset with a thousand samples or a thousand datasets containing

<sup>94</sup> With the occasional—and deeply appreciated—distraction of Syrielle Montariol on unrelated NLP projects (Montariol et al. 2022).

Marcheggiani and Titov, “Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations” *TACL* 2016

X. Hu et al., “SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction” *EMNLP* 2020

Soares et al., “Matching the Blanks: Distributional Similarity for Relation Learning” *ACL* 2019

6427 one sample each. Indeed, these models do not model the interactions  
6428 between samples.

6429

6430 *Aggregate methods* map a set of unsupervised samples to a set of facts  
6431 at once. There is not necessarily a direct correspondence between  
6432 extracted facts and samples in the dataset, even though most ag-  
6433 gregate models still provide a sentential prediction. In this setup, a  
6434 dataset containing a single sentence would be meaningless; it would  
6435 boil down to a sentential approach.

6436

6437 This distinction can also be made for unsupervised models, and indeed  
6438 Chapter 3 follows mostly a sentential approach, whereas Chapter 4 pur-  
6439 poses to introduce the aggregate approach to the unsupervised setting.

6440

6441 In Chapter 2, we also presented unsupervised relation extraction mod-  
6442 els. Unsupervised models need to rely on modeling hypotheses to capture  
6443 the notion of relation. While these hypotheses are not always clearly stated  
6444 in articles, they are central to the design of unsupervised approaches. For  
6445 our review, we decided to exhibit the key modeling hypotheses of relevant  
6446 models. Formalizing these hypotheses allows us to have a clear under-  
6447 standing of what kind of relations cannot be modeled by a given model.  
6448 Furthermore, it simplifies the usually challenging task of designing an un-  
6449 supervised relation extraction loss.

As a reminder, the modeling hypothe-  
ses are listed in Appendix B.

6449

#### 6450 **Regularizing discriminative approaches for deep encoders.**

6451 In Chapter 3, we introduced the first unsupervised model that does not  
6452 rely on hand-engineered features. In particular, we identified two criti-  
6453 cal weaknesses of previous discriminative models which hindered the use  
6454 of deep neural networks. These weaknesses relate to the model’s output,  
6455 which tends to collapse to a trivial—either deterministic or uniform—  
6456 distribution. We introduced two relation distribution losses to alleviate  
6457 these problems: a skewness loss pushes the prediction away from a uni-  
6458 form distribution, and a distribution distance loss prevents the output  
6459 from collapsing to a deterministic distribution. This allowed us to train  
6460 a PCNN model to cluster unsupervised samples in clusters conveying the  
6461 same relation.

6462

#### 6463 **Exploiting the dataset structure using graph-based models.**

6464 In Chapter 4, we investigated aggregate approaches for unsupervised re-  
6465 lation extraction. We encoded the relation extraction problem as a graph  
6466 labeling—or attributing—problem. We then showed that information can  
6467 be leveraged from this structure by probing distributional regularities of  
6468 random paths. To exploit this information, we designed an assumption us-  
6469 ing our experience from Chapter 2 to leverage the structure of the graph  
6470 to supervise a relation extraction model. We then proposed an approach  
6471 based on this hypothesis by modifying the Weisfeiler–Leman isomorphism  
6472 test to use a 1-Wasserstein distance.

6473

6474

6475 From a higher vantage point, we can say that we first assisted the  
6476 development of deep learning approaches for the task of unsupervised re-  
6477 lation extraction, and then helped open a new direction of research on  
6478 aggregate approaches in the unsupervised setup using graph-based mod-  
6479 els. Both of these research objects were somewhat natural developments  
6480 following current trends in machine learning research.

## Perspectives

**Using language modeling for relation extraction.** A recent trend in NLP has been to encode all tasks as language models. The main embodiment of this trend is T5 (Raffel et al. 2020). T5 is trained as a masked language model (MLM, Section 1.3.4.2) on a sizeable “common crawl” of the web. Then, it is fine-tuned by prefixing the sequence with a task-specific prompt such as “translate English to German:”. Relation extraction can also be trained as a text-to-text model in the supervised setup (Trisedya et al. 2019). Extending this model to the unsupervised setup—for example, through the creation of pseudo-labels—could allow us to leverage the large amount of linguistic information contained in the T5 parameters. In the same vein, Ushio et al. (2021) propose to use predefined and learned prompts for relation prediction, for example by filling in the following template: “Today, I finally discovered the relation between  $e_1$  and  $e_2$ :  $e_1$  is the <BLANK/> of  $e_2$ .”

More generally, relation extraction is closely related to language models. The first model we experimented on during this Ph.D. candidacy was a pre-trained language model used to fill sentences such as “The capital of Japan is <BLANK/>.” While Vaswani et al. (2017) was already published at the time, pre-trained transformer language models were not widely available yet. We used a basic LSTM, which was strongly biased in favor of entities often appearing in the dataset. In practice, the model predicted “London” as the capital of most small countries. However, as we showcased in Section 2.5.6, large transformer-based models such as BERT (Devlin et al. 2019) perform well out-of-the-box on unsupervised relation extraction. An additional argument in favor of transformer-based language models comes from Chapter 3. Indeed, the *fill-in-the-blank* model seeks to predict an entity blanked in the input; this is similar to the MLM task. More abstractly, language purposes to describe a reality which can be understood—among other things—through the concept of relation. And indeed, if one understands language, one must understand the relations conveyed by language. Using a model of language as a basis for a model of relations is promising, as long as the semantic fragment of language unrelated to relations can be discarded.

**Dataset-level modeling hypotheses.** In the past few years, graph-based approaches have gained traction in the information extraction field (Fu et al. 2019; Qian et al. 2019) and we can only expect this interest to continue growing in the future. While knowledge of the language should be sufficient to understand the relation underlying most samples, it is challenging to design an unsupervised loss solely relying on linguistic information. Furthermore, following distributional linguistics, language—and thus relations conveyed by language—are acquired through structured repetitions. The concept of repetition captured by graph adjacency can therefore also provide a theoretical basis for the design of modeling hypotheses. We can even argue that capturing the structure of the data is an ontologically prior modeling level. For this reason, we think that relation graphs should provide a better basis for the formulation of modeling hypotheses.

**Complex relations.** Several simplifying assumptions were made to define the relation extraction task. For example, we assume all relations to be binary, holding between exactly two entities. However,  $n$ -ary relations

Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer” JMLR 2020

The name T5 comes from “Text-To-Text Transfer Transformer” since it recasts every NLP task as a text-to-text problem.

Ushio et al., “Distilling Relation Embeddings from Pretrained Language Models” 2021

Vaswani et al., “Attention is All you Need” NeurIPS 2017

Qian et al., “GraphIE: A Graph-Based Framework for Information Extraction” 2019

6535 are needed to model complex interrelationships. For example, encoding  
6536 the fact that “a drug  $e_1$  can be used to treat a disease  $e_2$  when the patient  
6537 has genetic mutation  $e_3$ ” necessitates a ternary relation. This problem has  
6538 been tackled for a long time (McDonald et al. 2005; Song et al. 2018). The  
6539 graph-based approaches have a natural extension to  $n$ -ary relation in the  
6540 form of hypergraphs, which are graphs with  $n$ -ary edges. Since the hyper-  
6541 graph isomorphism problem can be polynomially reduced to the standard  
6542 graph isomorphism problem (Zemlyachenko et al. 1985), we can expect  
6543  $n$ -ary extension of graph-based relation extraction approaches to work as  
6544 well as standard relation extraction.

6545 A related problem is the one of fact qualification. The fact “Versailles  
6546 capital of France” only held until the 1789 revolution. In the Wikidata  
6547 parlance, these kinds of details are called *qualifiers*. In particular, the tem-  
6548 poral qualification can be critical to certain relation extraction datasets  
6549 (Jiang et al. 2019). Some information extraction datasets already include  
6550 this information (Mesquita et al. 2019); however, little work has been made  
6551 in this direction yet. Qualifiers could be generated from representations  
6552 of relations in a continuous manifold such as the one induced by a sim-  
6553 ilarity space for few-shot evaluation. However, learning to map relation  
6554 embeddings to qualifiers in an unsupervised fashion might prove difficult.

6555

6556

6557

6558

6559

6560

6561

6562

6563

6564

6565

6566

6567

6568

6569

6570

6571

6572

6573

6574

6575

6576

6577

6578

6579

6580

6581

6582

6583

6584

6585

6586

6587

6588

## Appendix A

### Résumé en français

**META-RÉSUMÉ** Détecter les relations exprimées dans un texte est un problème fondamental de la compréhension du langage naturel. Il constitue un pont entre deux approches historiquement distinctes de l'intelligence artificielle, celles à base de représentations symboliques et distribuées. Cependant, aborder ce problème sans supervision humaine pose plusieurs problèmes et les modèles non supervisés ont des difficultés à faire écho aux avancées des modèles supervisés. Cette thèse aborde deux lacunes des approches non supervisées : le problème de la régularisation des modèles discriminatifs et le problème d'exploitation des informations relationnelles à partir des structures des jeux de données. La première lacune découle de l'utilisation de réseaux neuronaux profonds. Ces modèles ont tendance à s'effondrer sans supervision. Pour éviter ce problème, nous introduisons deux fonctions de coût sur la distribution des relations pour contraindre le classifieur dans un état entraînable. La deuxième lacune découle du développement des approches au niveau des jeux de données. Nous montrons que les modèles non supervisés peuvent tirer parti d'informations issues de la structure des jeux de données, de manière encore plus décisive que les modèles supervisés. Nous exploitons ces structures en adaptant les méthodes non supervisées existantes pour capturer les informations topologiques à l'aide de réseaux convolutifs pour graphes. De plus, nous montrons que nous pouvons exploiter l'information mutuelle entre les données topologiques et linguistiques pour concevoir un nouveau paradigme d'entraînement pour l'extraction non supervisée de relations.

Le monde est doté d'une structure, qui nous permet de le comprendre. Cette structure est en premier lieu apparente à travers la répétition de nos expériences sensorielles. Parfois, nous voyons un chat, puis un autre chat. Les entités émergent de la répétition de l'expérience de *félinité* que nous avons ressentie. De temps en temps, nous pouvons également observer un chat à l'intérieur d'un carton ou une personne à l'intérieur d'une pièce. Les relations sont le mécanisme explicatif qui sous-tend ce deuxième type de répétition. Une relation régit une interaction entre au moins deux objets. Nous supposons qu'une relation à l'intérieur existe parce que nous avons vécu à plusieurs reprises la même interaction entre un conteneur et son contenu. Le vingtième siècle a été traversé par le développement du structuralisme, qui considérait que les interrelations entre phénomènes étaient plus éclairantes que l'étude des phénomènes pris isolément. En d'autres termes, nous pourrions mieux comprendre ce qu'est un chat en étudiant

“ *Puisque tu fais de la géométrie et de la trigonométrie, je vais te donner un problème : Un navire est en mer, il est parti de Boston chargé de coton, il jauge 200 tonneaux ; il fait voile vers le Havre, le grand mât est cassé, il y a un mousse sur le gaillard d'avant, les passagers sont au nombre de douze, le vent souffle N.-E.-E., l'horloge marque 3 heures un quart d'après-midi, on est au mois de mai... On demande l'âge du capitaine ?*

— Gustave Flaubert, « Lettre du 16 mai 1843 à sa sœur » (1926)

Flaubert se moque de l'enseignement mathématique à « son vieux rat » (Caroline Flaubert). Celle-ci ne répondit pas en prenant en compte la corrélation entre la responsabilité de diriger un navire jaugeant 200 tonneaux et l'avancée de la carrière du capitaine.

“ *À travers l'espace feuilleté des vingt-sept pairs, Faustroll évoqua vers la troisième dimension : De Baudelaire, le Silence d'Edgard Poë, en ayant soin de retraduire en grec la traduction de Baudelaire.*

— Alfred Jarry, *Gestes et opinions du docteur Faustroll* (1911)



Le chat du Cheshire de TENNIEL (1889) vous fournit une expérience de *félinité*.

6643 ses relations avec d'autres entités plutôt qu'en énumérant les caractéris-  
6644 tiques de notre expérience de la *félinité*. De ce point de vue, le concept de  
6645 relation est crucial dans notre compréhension du monde.

6646 Les langues naturelles saisissent la structure sous-jacente de ces répé-  
6647 titions à travers un processus que nous ne comprenons pas entièrement.  
6648 L'un des objectifs de l'intelligence artificielle, appelé compréhension du  
6649 langage naturel, est d'imiter ce processus à l'aide d'algorithmes. Puisque  
6650 ce but nous échappe encore, nous nous efforçons d'en modéliser seulement  
6651 des parties. Cette thèse, suivant la perspective structuraliste, se concentre  
6652 sur l'extraction des relations véhiculées par la langue naturelle. En suppo-  
6653 sant que la langue naturelle est représentative de la structure sous-jacente  
6654 des expériences sensorielles,<sup>95</sup> nous devrions être en mesure de capturer les  
6655 relations en exploitant uniquement les répétitions, c'est-à-dire de manière  
6656 non supervisée.

6657

6658

## 6659 A.1 Contexte

6660

6661 L'extraction de relations peut nous aider à mieux comprendre le fonction-  
6662 nement des langues. Par exemple, la question de savoir s'il est possible  
6663 d'apprendre une langue à partir d'une petite quantité de données reste  
6664 une question ouverte en linguistique. L'argument de la pauvreté du sti-  
6665 mulus affirme que les enfants ne devraient pas être capable d'acquérir des  
6666 compétences linguistiques en étant exposés à si peu de données.<sup>96</sup> Il s'agit  
6667 de l'un des principaux arguments en faveur de la théorie controversée de  
6668 la grammaire universelle. Capturer des relations à partir de rien d'autre  
6669 qu'un petit nombre d'expressions en langue naturelle serait un premier  
6670 pas vers la réfutation de l'argument de la pauvreté du stimulus.

6671 Ce type de motivation derrière le problème d'extraction de relations  
6672 cherche à avancer l'*épistémè*.<sup>97</sup> Cependant, la plupart des avancées sur  
6673 cette tâche découlent d'une recherche de *technè*.<sup>98</sup> L'objectif final est de  
6674 construire un système ayant des applications dans le monde réel. Dans  
6675 cette perspective, l'intelligence artificielle a pour but de remplacer ou d'as-  
6676 sister les humains dans des tâches spécifiques. La plupart des tâches né-  
6677 cessitent une certaine forme de connaissances techniques (par exemple,  
6678 le diagnostic médical nécessite la connaissance des relations entre symp-  
6679 tômes et maladies). Le principal vecteur de connaissances est le langage  
6680 (par exemple, à travers l'éducation). Ainsi, l'acquisition de connaissances  
6681 à partir d'énoncés en langue naturelle est un problème fondamental pour  
6682 les systèmes destinés à avoir des applications concrètes.

6683 ALEX et al. (2008) présentent une analyse de l'impact des systèmes  
6684 d'extraction de connaissances à partir de textes sur un problème concret.  
6685 Leur article montre que les annotateurs humains peuvent utiliser un sys-  
6686 tème d'apprentissage automatique pour mieux extraire un ensemble d'in-  
6687 teractions protéine-protéine de la littérature biomédicale. Il s'agit clai-  
6688 rement d'une recherche de *technè* : les interactions protéine-protéine ne  
6689 sont pas de nouvelles connaissances, elles sont déjà publiées ; cependant,  
6690 le système améliore le travail de l'opérateur humain.

6691 Cet exemple d'application est révélateur du problème plus vaste de  
6692 l'explosion informationnelle. La quantité d'informations publiées n'a cessé  
6693 de croître au cours des dernières décennies. L'apprentissage automatique  
6694 peut être utilisé pour filtrer ou agréger cette grande quantité de données.  
6695 Pour ce genre de tâches, l'objet d'intérêt n'est pas le texte en lui-même

Les relations — quoique dans un sens plus restreint — sont l'un des dix *prédicaments* d'Aristote, les catégories d'objets d'appréhension humaine (GRACIA et NEWTON 2016).





<sup>95</sup> Les répétitions d'expériences sensorielles et de mots n'ont pas à être nécessairement identiques. Nous ne nous préoccupons ici que de la possibilité de résoudre les références. Même si nos expériences d'arbres s'accompagnent généralement d'expériences d'écorces, les mots « arbre » et « écorce » ne cooccurrent pas aussi souvent dans des expressions en langue naturelle. Cependant, leur relation méronymique est intelligible à la fois par l'expérience d'arbres et, entre autres, par l'utilisation de la préposition « de » dans les mentions écrites d'écorces.

<sup>96</sup> Ce qui impliquerait qu'une partie de la maîtrise du langage est innée.

<sup>97</sup> Du grec ancien *ἐπιστήμη* : connaissance, savoir.

<sup>98</sup> Du grec ancien *τέχνη* : technique, art.

ALEX et al., "Assisted curation : does text mining really help ?" PSB 2008

6697 mais la sémantique véhiculée, sa signification. Une question se pose alors :  
 6698 comment définir la sémantique que l'on cherche à traiter? En effet, la  
 6699 définition du concept de « sens » fait l'objet de nombreuses discussions  
 6700 dans la communauté philosophique. Bien que certains sceptiques, comme  
 6701 Quine, ne reconnaissent pas le sens comme un concept essentiel, ils es-  
 6702 timent qu'une description minimale du sens devrait au moins englober la  
 6703 reconnaissance de la synonymie. Cela fait suite à la discussion ci-dessus  
 6704 sur la reconnaissance des répétitions : si  est une répétition de , nous  
 6705 devrions pouvoir dire que  et  sont synonymes. En pratique, cela im-  
 6706 plique que nous devrions être en mesure d'extraire des classes de formes  
 6707 linguistiques ayant la même signification ou le même référent — la diffé-  
 6708 rence entre les deux n'est pas pertinente pour notre problème.

6709 Bien que la discussion au sujet du sens soit essentielle pour définir la  
 6710 notion de relation qui nous intéresse, il est important de noter que nous  
 6711 travaillons sur la langue naturelle; nous voulons extraire des relations à  
 6712 partir de textes, et non de répétitions d'entités abstraites. Pourtant, la  
 6713 correspondance entre les significants linguistiques et leur signification n'est  
 6714 pas bijective. Nous pouvons distinguer deux types de désalignement entre  
 6715 les deux : soit deux expressions renvoient au même objet (synonymie), soit  
 6716 la même expression renvoie à des objets différents selon le contexte dans  
 6717 lequel elle apparaît (homonymie). La première variété de désalignement est  
 6718 la plus courante, surtout au niveau de la phrase. Par exemple, « Paris est  
 6719 la capitale de la France » et « la capitale de la France est Paris » véhiculent  
 6720 le même sens malgré des formes écrites et orales différentes. Au contraire,  
 6721 le second type est principalement visible au niveau des mots. Par exemple,  
 6722 la préposition « de » dans les phrases « frémir de peur » et « Bellérophon  
 6723 de Corinthe » traduit soit une relation *causé par* soit une relation *né à*.  
 6724 Pour distinguer ces deux utilisations de « de, » nous pouvons utiliser des  
 6725 identifiants de relation tels que P828 pour *causé par* et P19 pour *né à*. Un  
 6726 exemple avec des identifiants d'entités — qui ont pour but d'identifier de  
 6727 manière unique les concepts d'entité — est donné dans la marge.

6728 Alors que la discussion qui précède donne l'impression que tous les  
 6729 objets s'inscrivent parfaitement dans des concepts clairement définis, en  
 6730 pratique, c'est loin d'être le cas. Très tôt dans la littérature de la représen-  
 6731 tation des connaissances, BRACHMAN (1983) a remarqué la difficulté de  
 6732 définir clairement des relations apparemment simples telles que *instance de*  
 6733 (P31). Ce problème découle de l'hypothèse selon laquelle la synonymie est  
 6734 transitive et, par conséquent, induit des classes d'équivalence. Cette hypo-  
 6735 thèse est assez naturelle puisqu'elle s'applique déjà au lien entre le langage  
 6736 et ses références : même si deux chats peuvent être très différents l'un de  
 6737 l'autre, nous les regroupons sous le même signifiant. Cependant, la langue  
 6738 naturelle est flexible. Lorsque nous essayons de capturer l'entité « chat, »  
 6739 il n'est pas tout à fait clair si nous incluons « un chat avec le corps d'une  
 6740 tarte aux cerises » dans les expériences ordinaires de chat.<sup>99</sup> Pour contour-  
 6741 ner ce problème, certains travaux récents sur le problème d'extraction de  
 6742 relations (HAN et al. 2018) définissent la synonymie comme une associa-  
 6743 tion continue intransitive. Au lieu de regrouper les formes linguistiques  
 6744 dans des classes bien définies partageant un sens unique, ils extraient une  
 6745 fonction de similarité mesurant la ressemblance de deux objets.

6746  
 6747 Maintenant que nous avons conceptualisé notre problème, concentrons-  
 6748 nous sur l'approche technique que nous proposons. Tout d'abord, pour  
 6749 résumer, cette thèse se concentre sur l'extraction non supervisée de rela-  
 6750 tions à partir de textes.<sup>100</sup> Les relations étant des objets capturant les



Paris (Q162121) n'est ni la capitale de la France, ni le prince de Troie, c'est le genre de la parisette à quatre feuilles. La capitale de la France est Paris (Q90) et le prince de Troie, fils de Priam, Pâris (Q167646). Illustration tirée de REDOUTÉ (1802).

“ La signification, c'est ce que devient l'essence, une fois divorcée d'avec l'objet de la référence et remarquée au mot.

— Willard Van Orman Quine, “Main Trends in Recent Philosophy : Two Dogmas of Empiricism” (1951)  
 Traduction de LAUGIER (2004)

BRACHMAN, “What IS-A Is and Isn't : An Analysis of Taxonomic Links in Semantic Networks” Computer 1983

<sup>99</sup> Le lecteur qui décrirait une telle entité comme étant un chat est invité à remplacer diverses parties du corps de ce chat imaginaire par des aliments jusqu'à ce que cesse son expérience de félicité.

HAN et al., “FewRel : A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation” EMNLP 2018

<sup>100</sup> Nous utilisons le texte car il s'agit de l'expression la moins ambiguë et la plus facile à traiter de la langue.

6751 interactions entre les entités, notre tâche est de trouver la relation reliant  
6752 deux entités données dans un texte. Par exemple, dans les trois exemples  
6753 suivants où les entités sont soulignées :

6754 Megrez<sub>e<sub>1</sub></sub> est une étoile de la constellation circumpolaire nord  
6755 de la Grande Ourse<sub>e<sub>2</sub></sub>.

6757 Posidonios<sub>e<sub>1</sub></sub> était un philosophe, astronome, historien, ma-  
6758 thématicien et professeur grec originaire d'Apamée<sub>e<sub>2</sub></sub>.

6759 Hipparque<sub>e<sub>1</sub></sub> est né à Nicée<sub>e<sub>2</sub></sub>, et est probablement mort sur  
6760 l'île de Rhodes, en Grèce.

6761 nous souhaitons reconnaître que les deux dernières phrases véhiculent la  
6762 même relation — dans ce cas,  $e_1$  né à  $e_2$  (P19) — ou du moins, suivant la  
6763 discussion du paragraphe précédent sur la difficulté de définir des classes de  
6764 relations, nous voulons reconnaître que les relations exprimées par les deux  
6765 derniers échantillons sont plus proches l'une de l'autre que celle exprimée  
6766 par le premier échantillon. Nous avançons que cela peut être réalisé par des  
6767 algorithmes d'apprentissage automatique. En particulier, nous étudions  
6768 comment aborder cette tâche en utilisant l'apprentissage profond. Bien  
6769 que l'extraction de relations puisse être abordée comme un problème de  
6770 classification supervisée standard, l'étiquetage d'un jeu de données avec  
6771 des relations précises est une tâche fastidieuse, en particulier lorsque l'on  
6772 traite des documents techniques tels que la littérature biomédicale étudiée  
6773 par ALEX et al. (2008). Un autre problème fréquemment rencontré par les  
6774 annotateurs est la question de l'applicabilité d'une relation, par exemple,  
6775 l'expression « le père<sub>e<sub>2</sub></sub> fondateur du pays<sub>e<sub>1</sub></sub> » doit-elle être étiquetée avec  
6776 la relation *produit-producteur*?<sup>101</sup> Nous examinons maintenant comment  
6777 l'apprentissage profond est devenu la technique la plus prometteuse pour  
6778 s'attaquer aux problèmes de traitement de la langue naturelle.

6780 La matière première du problème d'extraction de relations est le lan-  
6781 gage. Le traitement automatique de la langue naturelle (TAL)<sup>102</sup> était déjà  
6782 une direction de recherche importante dans les premières années de l'intel-  
6783 ligence artificielle. On peut le voir du point de vue *épistémè* dans l'article  
6784 fondateur de TURING (1950). Cet article propose la maîtrise du langage  
6785 comme preuve d'intelligence, dans ce qui est maintenant connu sous le nom  
6786 de test de Turing. La langue était également un sujet d'intérêt pour des ob-  
6787 jectifs de *technè*. En janvier 1954, l'expérience de Georgetown-IBM tente  
6788 de démontrer la possibilité de traduire le russe en anglais à l'aide d'or-  
6789 dinateurs (DOSTERT 1955). L'expérience proposait de traduire soixante  
6790 phrases en utilisant un dictionnaire bilingue pour traduire individuelle-  
6791 ment les mots et six types de règles grammaticales pour les réorganiser.  
6792 Les premières expériences ont suscité beaucoup d'attentes, qui ont été sui-  
6793 vies d'une inévitable déception, entraînant un « hiver » durant lequel les  
6794 fonds attribués à la recherche en intelligence artificielle ont été restreints. Si  
6795 la traduction mot à mot est assez facile dans la plupart des cas, la traduc-  
6796 tion de phrases entières est beaucoup plus difficile. La mise à l'échelle de  
6797 l'ensemble des règles grammaticales dans l'expérience de Georgetown-IBM  
6798 s'est avérée impraticable. Cette limitation n'était pas d'ordre technique.  
6799 Avec l'amélioration des systèmes de calcul, davantage de règles auraient  
6800 pu facilement être codées. L'un des problèmes identifiés à l'époque était  
6801 celui de la compréhension du sens commun.<sup>103</sup> Pour traduire ou, plus gé-  
6802 néralement, traiter une phrase, il faut la comprendre dans le contexte du  
6803 monde dans lequel elle a été prononcée. De simples règles de réécriture ne  
6804 peuvent pas rendre compte de ce processus.<sup>104</sup> Pour pouvoir traiter des

Nous utilisons les identifiants Wikidata (<https://www.wikidata.org>) pour in-  
dicer les entités et les relations. Les identifiants des entités commencent  
par Q, tandis que les identifiants des relations commencent par P. Par  
exemple, Q35120 est une entité.



Ariane se réveille sur le rivage de Naxos où elle a été abandonnée, peinture mu-  
rale d'Herculanum dans la collection du BRITISH MUSEUM (100 av. n. è.-  
100 de n. è.). Le navire au loin peut être identifié comme étant le bateau  
de Thésée, pour l'instant. Selon le point de vue philosophique du lecteur  
(Q1050837), son identité en tant que bateau de Thésée pourrait ne pas per-  
durer.

<sup>101</sup> L'annotateur de ce morceau de phrase dans le jeu de données SemEval 2010 Task 8 a considéré qu'il expr-  
imait effectivement la relation *produit-producteur*. La difficulté d'appliquer  
précisément une définition est un argument supplémentaire en faveur des  
approches basées sur les fonctions de similarité par rapport aux approches  
de classification.

<sup>102</sup> *natural language processing* (NLP)

TURING, "Computing Machinery and Intelligence" *Mind* 1950

<sup>103</sup> *commonsense knowledge*

<sup>104</sup> Par ailleurs, la grammaire est encore un domaine de recherche ac-  
tif. Nous ne comprenons pas parfaite-  
ment la réalité sous-jacente captu-  
rée par la plupart des mots et sommes  
donc incapables d'écrire des règles for-  
melles complètes pour leurs usages.  
Par exemple, MARQUE-PUCHEU (2008)  
présente un article de linguistique trait-  
tant de l'utilisation des prépositions  
françaises « de » et « à. » C'est l'un des  
arguments en faveur des approches non  
supervisées ; en évitant d'étiqueter ma-  
nuellement les jeux de données, nous  
évitons la limite des connaissances des  
annotateurs humains.



6805 phrases entières, un changement de paradigme était nécessaire.

6806 Une première évolution a eu lieu dans les années 1990 avec l'avènement  
6807 des approches statistiques (S. ABNEY 1996). Ce changement peut être at-  
6808 tribué en partie à l'augmentation de la puissance de calcul, mais aussi à  
6809 l'abandon progressif de préceptes linguistique essentialistes au profit de  
6810 préceptes distributionnalistes.<sup>105</sup> Au lieu de s'appuyer sur des experts hu-  
6811 mains pour concevoir un ensemble de règles, les approches statistiques  
6812 exploitent les répétitions dans de grands corpus de textes pour déduire  
6813 ces règles automatiquement. Par conséquent, cette progression peut égale-  
6814 ment être considérée comme une transformation des modèles d'intelligence  
6815 artificielle symbolique vers des modèles statistiques. La tâche d'extraction  
6816 de relations a été formalisée à cette époque. Et si les premières approches  
6817 étaient basées sur des modèles symboliques utilisant des règles prédéfi-  
6818 nées, les méthodes statistiques sont rapidement devenues la norme après  
6819 les années 1990. Cependant, ces modèles statistiques reposaient toujours  
6820 sur des connaissances linguistiques. Les systèmes d'extraction de relations  
6821 étaient généralement divisés en une première phase d'extraction de caracté-  
6822 ristiques linguistiques spécifiées à la main et une seconde phase où une  
6823 relation était prédite à partir de ces caractéristiques à l'aide de modèles  
6824 statistiques peu profonds.

6825 Une deuxième évolution est survenue dans les années 2010 lorsque les  
6826 approches d'apprentissage profond ont effacé la séparation entre les phases  
6827 d'extraction de caractéristiques et de prédiction. Les modèles d'apprentis-  
6828 sage profond sont entraînés pour traiter directement les données brutes,  
6829 dans notre cas des extraits de texte. À cette fin, des réseaux de neurones  
6830 capables d'approcher n'importe quelle fonction sont utilisés. Cependant,  
6831 l'entraînement de ces modèles nécessite généralement de grandes quantités  
6832 de données étiquetées. Il s'agit d'un problème particulièrement important  
6833 pour nous puisque nous traitons un problème non supervisé. En tant que  
6834 technique la plus récente et la plus efficace, l'apprentissage profond est  
6835 un choix naturel pour s'attaquer à l'extraction de relations. Cependant,  
6836 ce choix s'accompagne de problématiques que nous essayons de résoudre  
6837 dans ce manuscrit.

6838

6839

6840

6841

6842

6843

6844

6845

6846

6847

6848

6849

6850

6851

6852

6853

6854

6855

6856

6857

6858

## A.2 Régularisation des modèles discriminatifs d'extraction non supervisée de relations

L'évolution des méthodes d'extraction de relations non supervisées suit de près celle des méthodes de TAL décrite ci-dessus. La première approche utilisant des techniques d'apprentissage profond a été celle de MARCHEGGIANI et TITOV (2016). Cependant, une partie de leur modèle reposait toujours sur des caractéristiques linguistiques extraites en amont. La raison pour laquelle cette extraction ne pouvait pas être faite automatiquement, comme c'est habituellement le cas en apprentissage profond, est étroitement liée à la nature non supervisée du problème. Notre première contribution est de proposer une technique permettant l'entraînement d'approches d'extraction non supervisée de relations par apprentissage profond.

Nous avons identifié deux problèmes critiques des modèles discriminants existant qui entravent l'utilisation de réseaux neuronaux profonds pour l'extraction de caractéristiques. Ces problèmes concernent la sortie

<sup>105</sup> Noam Chomsky, l'un des linguistes essentialistes les plus importants, considère que la manipulation de probabilités d'extraits de texte ne permet pas d'acquérir une meilleure compréhension du langage. Suite au succès des approches statistiques, il n'a reconnu qu'un accomplissement de *technè* et non d'*épistémè*. Pour une réponse à cette position, voir S. ABNEY (1996) et NORVIG (2011).

“Cheval blanc n'est pas cheval.  
— “Gongsun Longzi” Cha-  
pitre 2 (circa 300 AV. N. È.)

Un paradoxe bien connu de la philosophie chinoise illustrant la difficulté de définir clairement le sens véhiculé par la langue naturelle. Ce paradoxe peut être résolu en désambiguïsant le mot « cheval. » Fait-il référence à « l'ensemble de tous les chevaux » (la vision méréologique) ou à « la chevalité » (la vision platonicienne) ? L'interprétation méréologique a été célèbrement — et de manière controversée — introduite par HANSEN (1983), voir FRASER (2007) pour une discussion des premières vues ontologiques du langage en Chine.



Frontispice de la bibliothèque OuCui-Pienne par CHEVALIER (1990). Une autre façon de cuisiner avec les lettres.

「白馬非馬」

6859 du classifieur, qui a tendance à s’effondrer en une distribution triviale, soit  
 6860 déterministe, soit uniforme. Nous proposons d’introduire deux fonctions  
 6861 de coût sur la distribution des relations pour atténuer ces problèmes :  
 6862 une fonction d’asymétrie éloigne la prédiction d’une loi uniforme, et une  
 6863 distance de distributions empêche la sortie de s’effondrer vers une distribu-  
 6864 tion déterministe. Cela nous a permis d’entraîner un modèle PCNN (ZENG  
 6865 et al. 2015) pour regrouper les échantillons non supervisés en partitions<sup>106</sup>  
 6866 véhiculant la même relation.

6867 Notre approche se base sur le problème de remplissage de texte à trous :

6868 “Le sol<sub>e<sub>1</sub></sub> a été la monnaie du ?<sub>e<sub>2</sub></sub> entre 1863 et 1985.”  
 6869

6870 Pour pouvoir remplir cette phrase avec le mot manquant, il est nécessaire  
 6871 de comprendre la relation véhiculée. Nous utilisons cette tâche comme un  
 6872 substitut nous permettant d’identifier la sémantique relationnelle de la  
 6873 phrase. Étant donné une phrase  $s$  contenant deux entités  $e$  exprimant la  
 6874 relation  $r$ , nous modélisons la probabilité suivante :

$$6875 P(e_{-i} | s, e_i) = \sum_{r \in \mathcal{R}} \underbrace{P(r | s)}_{(i) \text{ classifieur}} \underbrace{P(e_{-i} | r, e_i)}_{(ii) \text{ prédicteur d'entité}} \quad \text{pour } i = 1, 2.$$

6877  
 6878 Nous utilisons un réseau profond (PCNN, ZENG et al. 2015) pour le clas-  
 6879 sifieur et le même modèle que MARCHEGGIANI et TITOV (2016) pour la  
 6880 prédiction d’entité. Le modèle résultant présente des instabilités, comme  
 6881 celle illustrée par la Figure A.1. Nous proposons deux fonctions de coût  
 6882 supplémentaires sur les paramètres  $\phi$  du classifieur pour résoudre ces pro-  
 6883 blèmes :

$$6884 \mathcal{L}_s(\phi) = \mathbb{E}_{(s,e) \sim \mathcal{U}(\mathcal{D})} [\text{H}(\text{R} | s, e; \phi)]$$

$$6885 \mathcal{L}_D(\phi) = \text{D}_{\text{KL}}(P(\text{R} | \phi) \| \mathcal{U}(\mathcal{R})).$$

6887  
 6888 La première fonction force la sortie du classifieur a avoir une entropie  
 6889 faible ce qui résout le problème de la Figure A.1. La seconde fonction s’as-  
 6890 sure qu’une variété de relations soient prédites pour différents échantillons.  
 6891 Ces deux fonctions nous permettent d’entraîner un réseau profond pour  
 6892 l’extraction non supervisée de relations comme le montrent les scores de  
 6893 la Table A.1.

6894

### 6895 A.3 Modélisation à l’aide de graphes de la 6896 structure des jeux de données 6897 6898 6899

6900 Comme mentionné dans la Section A.1, les approches récentes utilisent  
 6901 une définition plus souple des relations en extrayant une fonction de simi-  
 6902 larité au lieu d’un classifieur. De plus, elles considèrent un contexte plus  
 6903 large : au lieu de traiter chaque phrase individuellement, la cohérence glo-  
 6904 bale des relations extraites est prise en compte. Cependant, ce deuxième  
 6905 type d’approches a principalement été appliqué au cadre supervisé, avec  
 6906 une utilisation plus limitée dans le cadre non supervisé. Notre deuxième  
 6907 contribution concerne l’utilisation de ce contexte plus large pour l’extraction  
 6908 non supervisée de relations. En particulier, nous établissons des pa-  
 6909 rallèles avec le test d’isomorphisme de Weisfeiler–Leman pour concevoir  
 6910 de nouvelles méthodes utilisant conjointement des caractéristiques topo-  
 6911 logiques (au niveau des jeux de données) et linguistiques (au niveau des  
 6912 phrases).

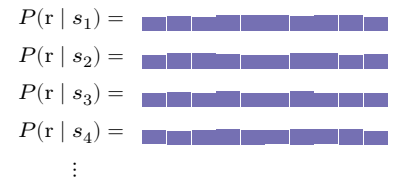
Cette section a fait l’objet d’une publi-  
 cation :

Étienne Simon, Vincent Guigue, Ben-  
 jamin Piwowarski. “Unsupervised In-  
 formation Extraction : Regularizing  
 Discriminative Approaches with Rela-  
 tion Distribution Losses” ACL 2019

ZENG et al., “Distant Supervision  
 for Relation Extraction via Piece-  
 wise Convolutional Neural Networks”  
 EMNLP 2015

<sup>106</sup> clusters

Distribution dégénérée :



Distribution désirée :

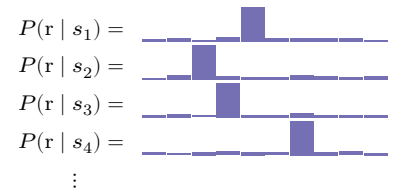


FIGURE A.1 : Illustration du problème d’uniformité. Le classifieur attribue la même probabilité à toutes les relations. À la place, nous souhaitons que le classifieur prédise clairement une relation unique pour chaque échantillon.

Modèle		B <sup>3</sup> F <sub>1</sub>
Classif.	Reg.	
Linear	$\mathcal{L}_{\text{VAE REG}}$	35,2
PCNN	$\mathcal{L}_{\text{VAE REG}}$	27,6
Linear	$\mathcal{L}_s + \mathcal{L}_D$	37,5
PCNN	$\mathcal{L}_s + \mathcal{L}_D$	<b>39,4</b>

TABLE A.1 : Résultats quantitatifs des méthodes de partitionnement sur le jeu de données NYT-FB. On distingue le classifieur utilisé (Classif.) de la régularisation utilisée (Reg.). La régularisation  $\mathcal{L}_{\text{VAE REG}}$  est celle issue de l’article de MARCHEGGIANI et TITOV (2016).

- 6913 Nous encodons le problème d'extraction de relations comme un problème d'étiquetage d'un multigraphe  $G = (\mathcal{E}, \mathcal{A}, \varepsilon, \rho, \varsigma)$  défini comme suit :
- 6914
- 6915 •  $\mathcal{E}$  est l'ensemble des nœuds qui correspondent aux entités.
  - 6916 •  $\mathcal{A}$  est l'ensemble des arcs qui connectent deux entités.
  - 6917 •  $\varepsilon_1 : \mathcal{A} \rightarrow \mathcal{E}$  associe à chaque arc son nœud d'origine (l'entité marquée  $e_1$ ),
  - 6918 •  $\varepsilon_2 : \mathcal{A} \rightarrow \mathcal{E}$  associe à chaque arc son nœud de destination (l'entité marquée  $e_2$ ),
  - 6919 •  $\varsigma : \mathcal{A} \rightarrow \mathcal{S}$  associe à chaque arc  $a \in \mathcal{A}$  la phrase correspondante contenant  $\varepsilon_1(a)$  et  $\varepsilon_2(a)$ ,
  - 6920 •  $\rho : \mathcal{A} \rightarrow \mathcal{R}$  associe à chaque arc  $a \in \mathcal{A}$  la relation entre les deux entités véhiculée par  $\varsigma(a)$ .

6921 Étant donné un chemin dans ce graphe :

$$6922 \quad e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3 \xrightarrow{r_3} e_4,$$

6923 nous avons conçu un algorithme de comptage basé sur l'exponentiation de la matrice d'adjacence de  $G$  et sur un échantillonnage préférentiel<sup>107</sup> qui nous permet d'approcher l'information mutuelle  $I(r_2; r_1, r_3) \approx 6,95$  bits. Elle se décompose en une entropie conditionnelle  $H(r_2 | r_1, r_3) \approx 1,06$  bits soustrait à l'entropie croisée<sup>108</sup>  $\mathbb{E}_{r_1, r_3} [H_{P(r_2)}(r_2 | r_1, r_3)] \approx 8,01$  bits. Cela signifie que la majeure partie de l'information relationnelle est extractible à partir du voisinage dans le graphe  $G$ .

6934 Fort de cette observation, nous utilisons l'hypothèse suivante pour concevoir un nouveau paradigme pour l'extraction non supervisée de relations :

6935 **Hypothèse distributionnelle faible sur le graphe d'extraction de relations.** Deux arcs véhiculent des relations similaires s'ils ont des voisinages similaires.

6936 Pour exploiter cette information de voisinage présente dans la topologie du multigraphe  $G$ , nous proposons de nous inspirer du test d'isomorphisme de Weisfeiler–Leman (WL, WEISFEILER et LEMAN 1968). Deux graphes sont dits isomorphes s'il existe un morphisme entre leur sommets qui conserve la relation de voisinage. Ce concept est illustré par la Figure A.2. Nous pouvons donc traduire l'hypothèse ci-dessus par l'affirmation que si les voisinages de deux échantillons sont isomorphes, alors ces deux échantillons véhiculent la même relation. Pour évaluer la proximité de deux voisinages, nous définissons  $\mathfrak{S}_{\rightarrow}(a, k)$ , le plongement par BERTcoder (voir Figure A.3) de la sphère de rayon  $k$  autour de l'arête  $a \in \mathcal{A}$  comme :

$$6942 \quad \begin{aligned} 6943 \quad \mathfrak{S}_{\rightarrow}(a, 0) &= \{a\} \\ 6944 \quad \mathfrak{S}_{\rightarrow}(a, k) &= \{x \in \mathcal{A} \mid \exists y \in \mathfrak{S}_{\rightarrow}(a, k-1) : \varepsilon_1(x) = \varepsilon_2(y)\} \\ 6945 \quad \mathfrak{S}_{\rightarrow}(a, k) &= \{\text{BERTcoder}(\varsigma(x)) \in \mathbb{R}^d \mid x \in \mathfrak{S}_{\rightarrow}(a, k)\}. \end{aligned}$$

6946 Ces sphères correspondent au voisinage à distance  $k$ . À partir de celles-ci, nous pouvons définir une fonction de distance prenant en compte le voisinage jusqu'à une distance  $K$  :

$$6947 \quad d(a, a'; \lambda) = \sum_{k=0}^K \frac{\lambda_k}{2} \sum_{\sigma \in \{\leftarrow, \rightarrow\}} W_1(\mathfrak{S}_{\sigma}(a, k), \mathfrak{S}_{\sigma}(a', k)),$$

6948 où  $W_1$  désigne la distance de Wasserstein d'ordre 1. En particulier, cette fonction évaluée en  $\lambda = [1]$  correspond à la distance habituelle entre plongements de phrases modulo l'utilisation de  $W_1$  à la place d'une distance

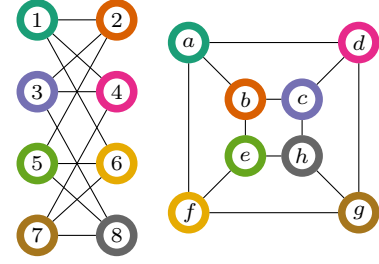


FIGURE A.2 : Exemple de graphes isomorphes. Chaque nœud  $i$  dans le graphe de gauche correspond à la  $i$ -ième lettre de l'alphabet dans le graphe de droite. Par ailleurs, ces graphes contiennent des automorphismes non-triviaux, par exemple en associant le nœud  $i$  au nœud  $9 - i$ .

<sup>107</sup> importance sampling

<sup>108</sup> cross-entropy

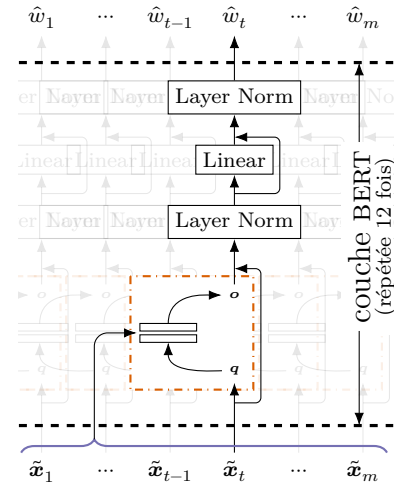


FIGURE A.3 : Schéma de BERT (DEVLIN et al. 2019), un modèle de langage masqué basé sur un transformer. Le modèle est entraîné à reconstruire des mots  $\hat{w}_t$  corrompus en  $\tilde{w}_t$  (plongés en  $\tilde{x}_t$ ). BERTcoder est une spécialisation de ce modèle pour l'extraction de relations (SOARES et al. 2019).

KIPF et WELLING (2017) ont déjà tracé un parallèle entre WL et les approches à base de réseaux neuronaux convolutifs pour graphes (GCN). Toutefois, nous avançons que les fonctions d'apprentissage habituellement utilisées pour les GCN ne sont pas adaptées au problème d'extraction non supervisée de relations.

6967 cosinus. Pour des raisons de limites de calcul, nous fixons  $K = 2$ . Dans ce  
 6968 cas,  $d(a_1, a_2, [1, 0]^T)$  correspond à la distance linguistique entre deux échan-  
 6969 tillons  $a_1, a_2 \in \mathcal{A}$ , tandis que  $d(a_1, a_2, [0, 1]^T)$  correspond à la distance to-  
 6970 pologique entre les voisinages des échantillons  $a_1$  et  $a_2$ . Nous proposons de  
 6971 faire coïncider ces deux distances pour tirer parti de l’information mutuelle  
 6972 au voisinage et à la phrase afin d’identifier la sémantique relationnelle des  
 6973 échantillons. Pour ce faire, nous introduisons une fonction de coût par  
 6974 triplet :<sup>109</sup>

$$\mathcal{L}_{\text{LT}}(a_1, a_2, a_3) = \max \begin{pmatrix} 0, \zeta + 2(d(a_1, a_2, [1, 0]^T) - d(a_1, a_2, [0, 1]^T))^2 \\ - (d(a_1, a_2, [1, 0]^T) - d(a_1, a_3, [0, 1]^T))^2 \\ - (d(a_1, a_3, [1, 0]^T) - d(a_1, a_2, [0, 1]^T))^2 \end{pmatrix}.$$

6980  
 6981 Des résultats préliminaires sur l’utilisation d’informations topologiques  
 6982 sont donnés dans la Table A.2. Comme on pouvait s’y attendre, l’infor-  
 6983 mation relationnelle encodée dans le voisinage d’ordre 1 du graphe est  
 6984 moindre que celle directement contenue dans la phrase. Toutefois, ces in-  
 6985 formations peuvent être combinées ce qui permet d’améliorer significati-  
 6986 vement la performance du modèle d’extraction de relation.

6987

6988

## 6989 A.4 Conclusion

6990

6991 Pendant ma candidature au doctorat, je me suis—principalement<sup>110</sup>—con-  
 6992 centré sur l’étude de l’extraction non supervisée de relations. Dans cette  
 6993 tâche, étant donné un ensemble de phrases et de paires d’entités, nous  
 6994 recherchons l’ensemble des faits véhiculés  $(e_1, r, e_2)$ , tels que  $r$  exprime la  
 6995 relation entre  $e_1$  et  $e_2$  dans un échantillon. Pour mener à bien cette tâche,  
 6996 nous avons suivi deux axes de recherche principaux : premièrement, la  
 6997 question de savoir comment entraîner un réseau neuronal profond pour  
 6998 l’extraction non supervisée de relations ; deuxièmement, la question de  
 6999 savoir comment tirer parti de la structure d’un ensemble de données pour  
 7000 obtenir des informations supplémentaires pour la tâche d’extraction de  
 7001 relations sans supervision.

7002 Plus grossièrement, nous avons d’abord aidé au développement d’ap-  
 7003 proches d’apprentissage profond pour la tâche d’extraction non supervisée  
 7004 de relations, puis contribué à ouvrir une nouvelle direction de recherche  
 7005 sur les approches au niveau des jeux de données dans la configuration non  
 7006 supervisée utilisant des modèles basés sur des graphes. Ces deux objets de  
 7007 recherche étaient en quelque sorte des développements naturels suivant les  
 7008 tendances actuelles de la recherche en apprentissage automatique.

7009

7010

7011

7012

7013

7014

7015

7016

7017

7018

7019

7020

<sup>109</sup> triplet loss

Modèle	Précision
Linguistique (BERT)	69,46
Topologique ( $W_1$ )	65,75
Tous les deux	72,18

TABLE A.2 : Résultats quantitatifs des méthodes à base de graphe sur le jeu de données FewRel (HAN et al. 2018). Ces résultats portent uniquement sur les échantillons de FewRel connectés par au moins une arête dans le graphe  $G$  du jeu de données T-REX.

<sup>110</sup> Avec la distraction occasionnelle—et profondément appréciée—de Syrielle Montariol sur d’autres projets de TAL (MONTARIOL et al. 2022).

## Appendix B

### List of Assumptions

Modeling hypotheses are central to relation extraction approaches, especially unsupervised ones (see Chapter 2). This appendix list all assumptions introduced in the previous chapters in alphabetical order, with reference to the section in which it was introduced, and whenever possible a counterexample exposing what kind of construct cannot be captured by making this hypothesis.

**Assumption  $\mathcal{H}_{1 \rightarrow 1}$ :** *All relations are one-to-one.*

1  $\rightarrow$  1

$\forall r \in \mathcal{R}: r \bullet \check{r} \cup \mathbf{I} = \check{r} \bullet r \cup \mathbf{I} = \mathbf{I}$

Appeared Section 2.5.6.

Counterexample: “Josetsu *born in* Kyushu” and “Minamoto no Shunrai *born in* Kyushu.”

**Assumption  $\mathcal{H}_{1\text{-ADJACENCY}}$ :** *There is no more than one relation linking any two entities.*

1-ADJACENCY

$\forall r_1, r_2 \in \mathcal{R}: r_1 \cap r_2 = \mathbf{0}$

Appeared Section 2.3.2.

Counterexample: “Khayyam *born in* Nishapur” and “Khayyam *died in* Nishapur.”

**Assumption  $\mathcal{H}_{1\text{-NEIGHBORHOOD}}$ :** *Two samples with the same neighborhood in the relation extraction graph convey the same relation.*

1-NEIGHBORHOOD

$\forall a, a' \in \mathcal{A}: \mathcal{N}(a) = \mathcal{N}(a') \implies \rho(a) = \rho(a')$

Appeared Section 4.4.3.

Counterexample: *born in* and *died in*. Since the arc-neighborhood  $\mathcal{N}$  is split between in-and out-neighborhood, this hypothesis is close to  $\mathcal{H}_{\text{TYPE}}$ .

The main difference being that the partitions (types) of  $\mathcal{H}_{\text{TYPE}}$  can't overlap. While a relation which can have any type as a subject can't be modeled under the  $\mathcal{H}_{\text{TYPE}}$  hypothesis, it will simply correspond to a distribution with mass on all entities in the  $\mathcal{H}_{1\text{-NEIGHBORHOOD}}$  assumption.

**Assumption  $\mathcal{H}_{\text{BICLIQUE}}$ :** *Given a relation, the entities are independent of one another:  $e_1 \perp e_2 \mid r$ . In other words, given a relation, all possible head entities are connected to all possible tail entities.*

BICLIQUE

$\forall r \in \mathcal{R}: \exists A, B \subseteq \mathcal{E}: r \bullet \check{r} = \mathbf{1}_A \wedge \check{r} \bullet r = \mathbf{1}_B$

7075 Appeared Section 2.5.4.

7076 Counterexample: most relations should infringe this assumption since it is  
7077 decomposable into two unary predicates: whether the entity is part of  $A$   
7078 and whether it is part of  $B$ . For example “Alonzo Church *died in Hudson*”  
7079 and “Alan Turing *died in Wilmslow*” are true but “Alonzo Church *died in*  
7080 *Wilmslow*” is false.

7081

7082 **Assumption**  $\mathcal{H}_{\text{BLANKABLE}}$ : *The relation can be predicted by the text sur-*  
7083 *rounding the two entities alone. Formally, using  $\text{blanked}(s)$  to designate*  
7084 *the tagged sentence  $s \in \mathcal{S}$  from which the entities surface forms were*  
7085 *removed, we can write:*

BLANKABLE

7086  $r \perp \mathbf{e} \mid \text{blanked}(s)$ .

7087

7088 Appeared Section 3.1.0.

7089 Counterexample: some surface forms are mapped to different relations  
7090 depending on the nature of the entities; in FewRel, “ $\underline{?}_{e_1}$  is part of  $\underline{?}_{e_2}$ ”  
7091 can both convey *part of* and *part of constellation*.

7092

7093

7094 **Assumption**  $\mathcal{H}_{\text{CTX}(1\text{-ADJACENCY})}$ : *Two samples with the same contextualized*  
7095 *representation of their entities’ surface forms convey the same relation.*

CTX(1-ADJACENCY)

7096  $\forall (s, \mathbf{e}, r), (s', \mathbf{e}', r') \in \mathcal{D}_{\mathcal{R}}$ :

7097  $\text{ctx}_1(s) = \text{ctx}_1(s') \wedge \text{ctx}_2(s) = \text{ctx}_2(s') \implies r = r'$

7098

7099 Appeared Section 2.5.7.

7100 Finding a counterexample for this assumption is quite difficult since it  
7101 depends on the operation performed by the contextualization function  
7102  $\text{ctx}$ . In this sense, it is a weak assumption.

7103

7104 **Assumption**  $\mathcal{H}_{\text{DISTANT}}$ : *A sentence conveys all the possible relations be-*  
7105 *tween all the entities it contains.*

DISTANT

7106  $\mathcal{D}_{\mathcal{R}} = \mathcal{D} \bowtie \mathcal{D}_{\text{KB}}$

7107 where  $\bowtie$  denotes the natural join operator:

7108

7109  $\mathcal{D} \bowtie \mathcal{D}_{\text{KB}} = \{(s, e_1, e_2, r) \mid (s, e_1, e_2) \in \mathcal{D} \wedge (e_1, e_2, r) \in \mathcal{D}_{\text{KB}}\}$ .

7110

7111 Appeared Section 2.2.2.

7112 Counterexample: “Chekhov found himself coughing blood, and in 1886 the  
7113 attacks worsened, but he would not admit his tuberculosis to his family  
7114 or his friends.” does not convey the fact “Anton Chekhov *cause of death*  
7115 *Tuberculosis*,” it only conveys “Anton Chekhov *has medical condition* Tu-  
7116 *berculosis*.”

7117

7118

7119 **Assumption**  $\mathcal{H}_{\text{MULTI-INSTANCE}}$ : *All facts  $(\mathbf{e}, r) \in \mathcal{D}_{\text{KB}}$  are conveyed by at*  
7120 *least one sentence of the unlabeled dataset  $\mathcal{D}$ .*

MULTI-INSTANCE

7121  $\forall (e_1, e_2, r) \in \mathcal{D}_{\text{KB}} : \exists (s, e_1, e_2) \in \mathcal{D} : (s, e_1, e_2)$  *conveys*  $e_1$  *r*  $e_2$

7122

7123 Appeared Section 2.4.2.

7124 Counterexample: Even though “Josetsu *born in Kyushu*” is present in  
7125 Wikidata, at the time of writing, this information is missing from its En-  
7126 glish Wikipedia page, thus an alignment of  $\mathcal{D} = \text{Wikipedia}$  with  $\mathcal{D}_{\text{KB}} =$   
7127 *Wikidata* would not verify  $\mathcal{H}_{\text{MULTI-INSTANCE}}$ .

7128

7129 **Assumption**  $\mathcal{H}_{\text{PULLBACK}}$ : *It is possible to find the relation conveyed by a* PULLBACK  
 7130 *sample by looking at the entities alone and ignoring the sentence; and*  
 7131 *conversely by looking at the sentence alone and ignoring the entities.*

7132  $\mathcal{D} = \mathcal{S} \times_{\mathcal{X}} \mathcal{E}^2$ .

7133

7134 Appeared Section 2.2.1.

7135 Entails  $\mathcal{H}_{1\text{-ADJACENCY}}$ .

7136 Counterexample: Unless the reader is familiar with biographies of early  
 7137 Chinese philosophers, the relation between Q1362266 “Gongsun Long” and  
 7138 Q197430 “Zhao” should not be immediately obvious.

7139

7140 **Assumption**  $\mathcal{H}_{\text{TYPE}}$ : *All entities have a unique type, and all relations are* TYPE  
 7141 *left and right restricted to one of these types.*

7142  $\exists \mathcal{T}$  partition of  $\mathcal{E} : \forall r \in \mathcal{R} : \exists X, Y \in \mathcal{T} : r \bullet \check{r} \cup \mathbf{1}_X = \mathbf{1}_X \wedge \check{r} \bullet r \cup \mathbf{1}_Y = \mathbf{1}_Y$

7143

7144 Appeared Section 2.5.3.

7145 Counterexample: “Deneb *part of* Summer Triangle” (type pair: star–con-  
 7146 stellation) and “Mitochondrion *part of* Cytoplasm” (type pair: organelle–  
 7147 cellular component).

7148

7149

7150 **Assumption**  $\mathcal{H}_{\text{UNIFORM}}$ : *All relations occur with equal frequency.* UNIFORM

7151  $\forall r \in \mathcal{R} : P(r) = \frac{1}{|\mathcal{R}|}$

7152

7153 Appeared Section 2.5.5.

7154 Counterexample: The relation “*worshipped by*” generally appears quite a  
 7155 lot less than “*place of burial*” whether measured through the number of  
 7156 facts in Wikidata or as the number of sentences conveying these relations  
 7157 in Wikipedia.

7158

7159

7160

7161

7162

7163

7164

7165

7166

7167

7168

7169

7170

7171

7172

7173

7174

7175

7176

7177

7178

7179

7180

7181

7182

7183  
7184  
7185  
7186  
7187  
7188  
7189  
7190  
7191  
7192  
7193  
7194  
7195  
7196  
7197  
7198  
7199  
7200  
7201  
7202  
7203  
7204  
7205  
7206  
7207  
7208  
7209  
7210  
7211  
7212  
7213  
7214  
7215  
7216  
7217  
7218  
7219  
7220  
7221  
7222  
7223  
7224  
7225  
7226  
7227  
7228  
7229  
7230  
7231  
7232  
7233  
7234  
7235  
7236



## Appendix C

### Datasets

In this appendix, we present the primary datasets used throughout this thesis. Each section corresponds to a dataset or group of datasets. We focus on the peculiarities which make each dataset unique and provide some statistics relevant to our task.

#### C.1 ACE

Automatic content extraction (ACE) is a NIST program that developed several datasets for the evaluation of entity chunking and relation extraction. It is the spiritual successor of MUC (Section C.4). In their nomenclature, the task of relation extraction is called relation detection and categorization (RDC). Datasets for relation extraction were released yearly between 2002 and 2005.<sup>111</sup> This makes comparison difficult; for example, in Chapter 2, we mention an ACE dataset for several models (Sections 2.3.4, 2.3.5, 2.4.1 and 2.4.5); however, the versions of the datasets differs.

A peculiarity of the ACE dataset is its hierarchy of relations. For example, the ACE-2003 dataset contains a *social* relation type, which is divided into several relation subtypes such as *grandparent* and *sibling*. Results can be reported either on the relation types or subtypes, usually using an  $F_1$  measure or a custom metric designed by ACE (Doddington et al. 2004) to handle directionality and the “*other*” relation (Section 2.1.1.1).

#### C.2 FewRel

FewRel (Han et al. 2018) is a few-shot relation extraction dataset. Given a query and several candidates, the model must decide which candidate conveys the relation closest to the one conveyed by the query. Therefore, FewRel is used to evaluate continuous relation representations; it is not typically used to evaluate a clustering model. For details on the few-shot setup, refer to Section 2.5.1.2.

The dataset was first constructed by aligning Wikipedia with Wikidata (Section C.8) using distant supervision (Section 2.2.2). Human annotators then hand-labeled the samples. The resulting dataset is perfectly balanced; all relations are represented by precisely 700 samples. The set of the 100 most common relations with good inter-annotator agreement was then

<sup>111</sup> The dataset from September 2002 is called ACE-2. This refers to the “second phase” of ACE. The pilot and first phase corpora only dealt with entity detection.

Doddington et al., “The automatic content extraction (ACE) program-tasks, data, and evaluation.” LREC 2004

Han et al., “FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation” EMNLP 2018

7291 divided into three splits, whose sizes are given in Table C.1. Since com-  
 7292 mon relations were strongly undersampled to obtain a balanced dataset,  
 7293 entities do not repeat much. The attributed multigraph (Section 4.1) cor-  
 7294 responding to the train split of FewRel is composed of several connected  
 7295 components. The larger one covers approximately 21% of the vertices,  
 7296 while more than half of all vertices are in connected components of size  
 7297 three or less.

7298 FewRel can be used for  $n$  way  $k$  shot evaluation, where usually  $n \in$   
 7299  $\{5, 10\}$  and  $k \in \{1, 5\}$ . For reference, Han et al. (2018) provides human  
 7300 performance on 5 way 1 shot (92.22% accuracy) and 10 way 1 shot (85.88%  
 7301 accuracy).

7302 A subsequent dataset released by the same team called FewRel 2.0  
 7303 (Gao et al. 2019) revisited the task by adding two variations:

7304 **Domain adaptation**, the training set of the original FewRel is used  
 7305 (Wikipedia–Wikidata), but the model is evaluated on biomedical  
 7306 literature (PubMed–UMLS) containing relations such as *may treat*  
 7307 and *manifestation of*.

7308 **Detecting other relation**, also called none-of-the-above, when the re-  
 7309 lation conveyed by the query does not appear in the candidates.

7310 While domain adaptation is an interesting problem, for unsupervised ap-  
 7311 proaches, the detection of *other* seems to defeat the point of modeling  
 7312 a similarity space instead of clustering relations. Furthermore, we only  
 7313 use FewRel as an evaluation tool and never train on it; using this second  
 7314 dataset made, therefore, little sense.

7315

7316

7317

7318

### 7319 C.3 Freebase

7320

7321

7322

7323

7324

7325

7326

7327

7328

7329

7330

7331

7332

7333

7334

7335

7336

7337

7338

7339

7340

7341

7342

7343

7344

Freebase (Bollacker et al. 2008) is a knowledge base (Section 1.4) started  
 in 2007 and discontinued in 2016. As one of the first widely available  
 knowledge bases containing general knowledge, Freebase was widely used  
 for weak supervision. In particular, it is the knowledge base used in the  
 original distant supervision article (Mintz et al. 2009). Freebase was a  
 collaborative knowledge base; as such, its content evolved through its ex-  
 istence. Therefore, even though Mintz et al. (2009), Yao et al. (2011) and  
 Marcheggiani and Titov (2016) all run experiments on Freebase, their re-  
 sults are not comparable since they use different versions of the dataset.  
 Data dumps are still provided by Google (2016); however, most of the  
 facts were transferred to the Wikidata knowledge base (Section C.8). Some  
 statistics about the latest version of Freebase are provided in Table C.2.  
 However, note that most relations in Freebase are scarcely used; only 6 760  
 relations appear in more than 100 facts. Furthermore, the concept of enti-  
 ties is quite wide in Freebase, in particular it makes use of a concept called  
 mediator (Chah 2017):

```

/m/02mjmr /topic/notable_for /g/125920
/g/125920 /c.../notable_for/object /gov.../us_president
/g/125920 /c.../notable_for/predicate /type/object/type

```

Here /m/02mjmr refers to “Barack Obama,” while /g/125920 is the me-  
 diator entity which is used to group together several statements about  
 /m/02mjmr.

Split	Relations	Samples
Train	64	44 800
Valid	16	11 200
Test	20	14 000

Table C.1: Statistics of the FewRel dataset. The test relations and samples are not publicly available.

Gao et al., “FewRel 2.0: Towards More Challenging Few-Shot Relation Classification” EMNLP 2019

Bollacker et al., “Freebase: a collaboratively created graph database for structuring human knowledge” SIGMOD 2008

Object	Number
Facts	3.1 billion
Entities	195 million
Relations	784 977

Table C.2: Statistics of the Freebase knowledge base at the time of its termination. Most relations (around 81%) appear only once in the knowledge base.

## 7345 C.4 MUC-7 TR

7346  
7347 The message understanding conferences (MUC) were organized by DARPA  
7348 in the 1980s and 1990s. The seventh—and last—conference (Chinchor  
7349 1998) introduced a relation extraction task called “template relation” (TR).  
7350 Three relations needed to be extracted: *employee of*, *location of* and *prod-*  
7351 *uct of*. Both the train set and evaluation set contained 100 articles. The  
7352 task was very much still in the “template filling” mindset; this can be seen  
7353 by the following example of extracted fact:  
7354

```
7355 <EMPLOYEE_OF-9602040136-5> :=
7356     PERSON: <ENTITY-9602040136-11>
7357     ORGANIZATION: <ENTITY-9602040136-1>
7358
7359 <ENTITY-9602040136-11> :=
7360     ENT_NAME: "Dennis Gillespie"
7361     ENT_TYPE: PERSON
7362     ENT_DESCRIPTOR: "Capt."
7363     / "the commander of Carrier Air Wing 11"
7364     ENT_CATEGORY: PER_MIL
7365
7366 <ENTITY-9602040136-1> :=
7367     ENT_NAME: "NAVY"
7368     ENT_TYPE: ORGANIZATION
7369     ENT_CATEGORY: ORG_GOVT
```

Chinchor, “Overview of MUC-7” MUC 1998

## 7371 C.5 New York Times

7372  
7373 The New York Times Annotated Corpus (NYT, Sandhaus 2008) was widely  
7374 used for relation extraction. The full dataset contains 1.8 million articles  
7375 from 1987 to 2007; however, smaller—and sadly, different—subsets are in  
7376 use. The subset we use in Chapter 3 was first extracted by Marcheggiani  
7377 and Titov (2016) and is supposed to be similar—but not identical—to the  
7378 one of Yao et al. (2011). This NYT subset only contains articles from 2000  
7379 to 2007 from which “noisy documents” were filtered out. Semi-structured  
7380 information such as tables and lists were also removed. The version of the  
7381 dataset we received from Diego Marcheggiani was already preprocessed,  
7382 with features listed in Section 3.3.2 already extracted.

Sandhaus, “The New York Times Annotated Corpus” LDC 2008

Marcheggiani and Titov, “Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations” TACL 2016

7383 The original dataset can be obtained from the following website:

```
7384     https://catalog.ldc.upenn.edu/LDC2008T19
```

7385  
7386  
7387 At the time of writing, once the license fee is paid, the only way to  
7388 obtain the subset of Marcheggiani and Titov (2016) and Chapter 3 is  
7389 through someone with access to this specific subset. This burdensome—  
7390 and expensive—procedure is one of the reasons for which we introduced  
7391 T-REx-based alternatives in Chapter 3.

## 7394 C.6 SemEval 2010 Task 8

7395  
7396  
7397 SemEval is the international workshop on semantic evaluation, which was  
7398 started in 1998 (then called Senseval) with the goal of emulating the

7399 message understanding conferences (Section C.4). In 2010, eighteen dif-  
 7400 ferent tasks were evaluated. Task number 8 was relation extraction. Sem-  
 7401 Eval 2010 Task 8 (Hendrickx et al. 2010) therefore refers to the dataset  
 7402 provided at the time of this challenge. It is a supervised relation extrac-  
 7403 tion dataset without entity linking and with non-unique entity reference  
 7404 (Section 2.1.2). Its statistics are listed in Table C.3. All samples were hand-  
 7405 labeled by human annotators with one of 19 relations. These 19 relations  
 7406 are built from 9 base relations, which can appear in both directions (Sec-  
 7407 tion 2.1.1.3), plus the *other* relation (Section 2.1.1.1). The 9 base relations  
 7408 in the dataset are:

- 7409 • *cause-effect*
- 7410 • *instrument-agency*
- 7411 • *product-producer*
- 7412 • *content-container*
- 7413 • *entity-origin*
- 7414 • *entity-destination*
- 7415 • *component-whole*
- 7416 • *member-collection*
- 7417 • *message-topic*

7418 SemEval 2010 Task 8 introduced an extensive evaluation system, most of  
 7419 which is described in Section 2.3.1. In particular, the official score of the  
 7420 competition was the half-directed macro- $\overline{F}_1$  (described in Section 2.3.1)  
 7421 which was referred to as “9 + 1-way evaluation taking directionality into  
 7422 account.”

## 7425 C.7 T-REX

7427 T-REX (Elsahar et al. 2018) is an alignment of Wikipedia with Wikidata.  
 7428 In particular, T-REX uses DBpedia abstracts (Brümmer et al. 2016), that  
 7429 is, the introductory paragraphs of Wikipedia’s articles. Its statistics are  
 7430 listed in Table C.4.

7431 In the final dataset, entities are linked using the DBpedia spotlight  
 7432 entity linker (Mendes et al. 2011). Furthermore, indirect entity links are  
 7433 extracted using coreference resolution and a “NoSub Aligner,” which as-  
 7434 sumes that the title of the article is implicitly mentioned by all sen-  
 7435 tences. Finally, some sequences of words are also linked to relations us-  
 7436 ing exact matches of Wikidata relation names. Both the datasets used in  
 7437 Chapters 3 and 4 only consider entities extracted by the spotlight entity  
 7438 linker (tagged `Wikidata_Spotlight_Entity_Linker`). The two datasets  
 7439 of Chapter 3 were filtered based on the tag of the predicate. SPO only  
 7440 contains samples whose predicate’s surface form appears in the sentence  
 7441 (tagged `Wikidata_Property_Linker`), while DS contains all samples with  
 7442 the two entities occurring in the same sentence (in other words, all samples  
 7443 except those tagged `NoSubject-Triple-aligner`).

## 7446 C.8 Wikidata

7449 Wikidata (Vrandečić and Krötzsch 2014) is a knowledge base (Section 1.4)  
 7450 started in 2012. Similar to the other projects of the Wikimedia Foundation,  
 7451 it is a collaborative enterprise; everyone can contribute new facts and  
 7452 entities. The introduction of new relations is made through the consensus

Hendrickx et al., “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals” SemEval 2010

Object	Number
Train samples	8 000
Test samples	2 717
Relations	$2 \times 9 + 1 = 19$

Table C.3: Statistics of the SemEval 2010 Task 8 dataset.

Elsahar et al., “T-REX: A Large Scale Alignment of Natural Language with Knowledge Base Triples” LREC 2018

Object	Number
Articles	3 million
Sentences	6.2 million
Facts	11 million
Relations	642

Table C.4: Statistics of the T-REX dataset.

Vrandečić and Krötzsch, “Wikidata: A Free Collaborative Knowledgebase” CACM 2014

7453  
7454  
7455  
7456  
7457  
7458  
7459  
7460  
7461  
7462  
7463  
7464  
7465  
7466  
7467  
7468  
7469  
7470  
7471  
7472  
7473  
7474  
7475  
7476  
7477  
7478  
7479  
7480  
7481  
7482  
7483  
7484  
7485  
7486  
7487  
7488  
7489  
7490  
7491  
7492  
7493  
7494  
7495  
7496  
7497  
7498  
7499  
7500  
7501  
7502  
7503  
7504  
7505  
7506

**Douglas Adams (Q42)** — relation (“ $e_1$ ”)

English writer and humorist  
Douglas Noël Adams | Douglas Noel Adams

**Statements**

*educated at* (P69) — relation (“ $r$ ”)

- St John’s College (Q691283) — object (“ $e_2$ ”)

**qualifiers** {

- start time* (P580) 1971
- end time* (P582) 1974
- academic major* (P812) English literature (Q186579)
- academic degree* (P512) Bachelor of Arts (Q1765120)

• Brentwood School (Q4961791) — object (“ $e_2$ ”)

**qualifiers** {

- start time* (P580) 1959
- end time* (P582) 1970

*work location* (P937) — relation (“ $r$ ”)

- London (Q84) — object (“ $e_2$ ”)

...

Figure C.1: Structure of a Wikidata page. Facts related to two relations are shown (“statement groups” in Wikidata parlance). This page can be translated into three  $\mathcal{E}^2 \times \mathcal{R}$  facts; the first has four additional qualifiers and the second has two additional qualifiers.

of long-term contributors to avoid the explosion of relations types observed on Freebase (section C.3).

Contrary to the way knowledge bases are presented in Section 1.4, Wikidata is not structured as a set of  $\mathcal{E}^2 \times \mathcal{R}$  triplets. Instead, in Wikidata, all entities have a page that lists facts of which the entity is the subject. These constitute our set  $\mathcal{D}_{\text{KB}} \subseteq \mathcal{E}^2 \times \mathcal{R}$ . Furthermore, Wikidata facts can be qualified by additional  $\mathcal{R} \times \mathcal{E}$  pairs. For example, Douglas Adams was *educated at* St John’s College *until* 1974. This structure is illustrated in Figure C.1. To be more precise, Wikidata could be modeled as a set of qualified facts, where a qualified fact is an element of  $\mathcal{E}^2 \times \mathcal{R} \times 2^{\mathcal{R} \times \mathcal{E}}$ .

7507  
7508  
7509  
7510  
7511  
7512  
7513  
7514  
7515  
7516  
7517  
7518  
7519  
7520  
7521  
7522  
7523  
7524  
7525  
7526  
7527  
7528  
7529  
7530  
7531  
7532  
7533  
7534  
7535  
7536  
7537  
7538  
7539  
7540  
7541  
7542  
7543  
7544  
7545  
7546  
7547  
7548  
7549  
7550  
7551  
7552  
7553  
7554  
7555  
7556  
7557  
7558  
7559  
7560

## Bibliography

- 7561  
7562  
7563  
7564  
7565  
7566  
7567  
7568  
7569  
7570  
7571  
7572  
7573 Abney, Steven (1996). “Statistical methods and linguistics”. In: *The balancing act: Combining symbolic and*  
7574 *statistical approaches to language*, pp. 1–26.
- 7575 Abney, Steven P. (1991). “Parsing by chunks”. In: *Principle-based parsing*. Springer, pp. 257–278.
- 7576 Agichtein, Eugene and Luis Gravano (2000). “Snowball: Extracting Relations from Large Plain-Text Col-  
7577 lections”. In: *Proceedings of the Fifth ACM Conference on Digital Libraries*. San Antonio, Texas, USA:  
7578 Association for Computing Machinery, pp. 85–94. ISBN: 158113231X. DOI: 10.1145/336597.336644. URL:  
7579 <https://dl.acm.org/doi/pdf/10.1145/336597.336644>.
- 7580 Alex, Beatrice, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roe-  
7581 buck, Richard Tobin, and Xinglong Wang (2008). “Assisted curation: does text mining really help?” In:  
7582 *Pacific Symposium on Biocomputing*. Vol. 13, pp. 556–567. URL: <https://psb.stanford.edu/psb-onlin>  
7583 [e/proceedings/psb08/alex.pdf](https://psb.stanford.edu/psb-online/proceedings/psb08/alex.pdf).
- 7584 Aone, Chinatsu, Lauren Halverson, Tom Hampton, and Mila Ramos-Santacruz (1998). “SRA: Description of  
7585 the IE<sup>2</sup> System Used for MUC-7”. In: *Seventh Message Understanding Conference (MUC-7): Proceedings of*  
7586 *a Conference Held in Fairfax, Virginia, April 29 – May 1, 1998*. URL: [https://aclanthology.org/M98-](https://aclanthology.org/M98-1012)  
7587 [1012](https://aclanthology.org/M98-1012).
- 7588 Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives (Nov.  
7589 2008). “DBpedia: A Nucleus for a Web of Open Data”. In: *Proceedings of 6th International Semantic Web*  
7590 *Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pp. 722–735. DOI: 10.1007/978-3-  
7591 -540-76298-0\_52. URL: <http://iswc2007.semanticweb.org/papers/715.pdf>.
- 7592 Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey Hinton (2016). “Layer Normalization”. arXiv: 1607.06450  
7593 [stat.ML].
- 7594 Babai, László (2015). “Graph Isomorphism in Quasipolynomial Time”. arXiv: 1512.03547 [cs.DS].
- 7595 — (2016). “Graph Isomorphism in Quasipolynomial Time”. arXiv: 1512.03547 [cs.DS].
- 7596 Bagga, Amit and Breck Baldwin (Aug. 1998). “Entity-Based Cross-Document Coreferencing Using the Vector  
7597 Space Model”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th Inter-*  
7598 *national Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for  
7599 Computational Linguistics, pp. 79–85. DOI: 10.3115/980845.980859. URL: [https://aclanthology.org](https://aclanthology.org/P98-1012)  
7600 [/P98-1012](https://aclanthology.org/P98-1012).
- 7601 Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly  
7602 Learning to Align and Translate”. In: *3rd International Conference on Learning Representations (ICLR),*  
7603 *Conference Track Proceedings* (May 7–9, 2015). Ed. by Yoshua Bengio and Yann LeCun. San Diego, CA,  
7604 USA. URL: <http://arxiv.org/abs/1409.0473>.
- 7605 Banko, Michele, Michael Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni (2007). “Open  
7606 Information Extraction from the Web”. In: *Proceedings of the 20th International Joint Conference on*  
7607 *Artificial Intelligence*. Hyderabad, India: Morgan Kaufmann Publishers Inc., pp. 2670–2676. URL: [https:](https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-429.pdf)  
7608 [//www.aaai.org/Papers/IJCAI/2007/IJCAI07-429.pdf](https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-429.pdf).
- 7609 Barrault, Loïc, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Gra-  
7610 ham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, et al. (Nov. 2020). “Findings of the 2020  
7611 Conference on Machine Translation (WMT20)”. In: *Proceedings of the Fifth Conference on Machine Trans-*  
7612 *lation*. Online: Association for Computational Linguistics, pp. 1–55. URL: [https://aclanthology.org/2](https://aclanthology.org/2020.wmt-1.1)  
7613 [020.wmt-1.1](https://aclanthology.org/2020.wmt-1.1).
- 7614 Beckett, Samuel (1955). *Molloy*.

- 7615 Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin (Mar. 2003). “A Neural Probabilistic  
7616 Language Model”. In: *The Journal of Machine Learning Research* 3, pp. 1137–1155. URL: [https://www.j](https://www.jmlr.org/papers/volume3/tmp/bengio03a.pdf)  
7617 [mlr.org/papers/volume3/tmp/bengio03a.pdf](https://www.jmlr.org/papers/volume3/tmp/bengio03a.pdf).
- 7618 Berant, Jonathan, Andrew Chou, Roy Frostig, and Percy Liang (Oct. 2013). “Semantic Parsing on Freebase  
7619 from Question-Answer Pairs”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural*  
7620 *Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1533–  
7621 1544. URL: <https://aclanthology.org/D13-1160>.
- 7622 Berners-Lee, Tim (1999). *Weaving the Web: The original design and ultimate destiny of the World Wide Web*  
7623 *by its inventor*. Harper San Francisco.
- 7624 Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). “Enriching Word Vectors with  
7625 Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.  
7626 DOI: 10.1162/tac1\_a\_00051. URL: <https://www.aclweb.org/anthology/Q17-1010>.
- 7627 Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). “Freebase: a collab-  
7628 oratively created graph database for structuring human knowledge”. In: *SIGMOD '08: Proceedings of the*  
7629 *2008 ACM SIGMOD international conference on Management of data*. Vancouver, Canada: Association for  
7630 Computing Machinery, pp. 1247–1250. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376746. URL:  
7631 <https://dl.acm.org/doi/pdf/10.1145/1376616.1376746>.
- 7632 Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko (2013).  
7633 “Translating Embeddings for Modeling Multi-relational Data”. In: *Advances in Neural Information Pro-*  
7634 *cessing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger.  
7635 Vol. 26. Curran Associates, Inc. URL: [https://proceedings.neurips.cc/paper/2013/file/1cecc7a77](https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf)  
7636 [928ca8133fa24680a88d2f9-Paper.pdf](https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf).
- 7637 Boulanger, Auguste (1897). “Contribution à l’étude des équations différentielles linéaires et homogènes inté-  
7638 grables algébriquement”. Thèses de doctorat.
- 7639 Brachman, Ronald (Oct. 1983). “What IS-A Is and Isn’t: An Analysis of Taxonomic Links in Semantic Net-  
7640 works”. In: *Computer* 16.10, pp. 30–36. ISSN: 1558-0814. DOI: 10.1109/MC.1983.1654194. URL: [https:](https://doi.ieeecomputersociety.org/10.1109/MC.1983.1654194)  
7641 [//doi.ieeecomputersociety.org/10.1109/MC.1983.1654194](https://doi.ieeecomputersociety.org/10.1109/MC.1983.1654194).
- 7642 Brin, Sergey (1999). “Extracting Patterns and Relations from the World Wide Web”. In: *The World Wide*  
7643 *Web and Databases*. Ed. by Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca. Berlin, Heidelberg:  
7644 Springer Berlin Heidelberg, pp. 172–183. ISBN: 978-3-540-48909-2. URL: [http://ilpubs.stanford.edu:8](http://ilpubs.stanford.edu:8090/421/1/1999-65.pdf)  
7645 [090/421/1/1999-65.pdf](http://ilpubs.stanford.edu:8090/421/1/1999-65.pdf).
- 7646 Brümmer, Martin, Milan Dojchinovski, and Sebastian Hellmann (May 2016). “DBpedia Abstracts: A Large-  
7647 Scale, Open, Multilingual NLP Training Corpus”. In: *Proceedings of the Tenth International Conference on*  
7648 *Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Associ-  
7649 ation (ELRA), pp. 3339–3343. URL: <https://aclanthology.org/L16-1532>.
- 7650 Bruna, Joan, Wojciech Zaremba, Arthur D. Szlam, and Yann LeCun (2014). “Spectral Networks and Locally  
7651 Connected Networks on Graphs”. In: *2nd International Conference on Learning Representations, ICLR*  
7652 *2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and  
7653 Yann LeCun. URL: <http://arxiv.org/abs/1312.6203>.
- 7654 Cai, Jin-Yi, Martin Fürer, and Neil Immerman (1992). “An optimal lower bound on the number of variables  
7655 for graph identification”. In: *Combinatorica* 12.4, pp. 389–410. URL: [https://people.cs.umass.edu/~im-](https://people.cs.umass.edu/~immerman/pub/opt.pdf)  
7656 [merman/pub/opt.pdf](https://people.cs.umass.edu/~immerman/pub/opt.pdf).
- 7657 Callison-Burch, Chris, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan  
7658 (July 2010). “Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for  
7659 Machine Translation”. In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and*  
7660 *MetricsMATR*. Uppsala, Sweden: Association for Computational Linguistics, pp. 17–53. URL: [https://ac-](https://aclanthology.org/W10-1703)  
7661 [lanthology.org/W10-1703](https://aclanthology.org/W10-1703).
- 7662 Ceglowski, Maciej (2014). *Web Design: The First 100 Years*. URL: [https://idlewords.com/talks/web\\_des-](https://idlewords.com/talks/web_design_first_100_years.htm)  
7663 [ign\\_first\\_100\\_years.htm](https://idlewords.com/talks/web_design_first_100_years.htm).
- 7664 Chah, Niel (2017). “Freebase-triples: A Methodology for Processing the Freebase Data Dumps”. arXiv: 1712  
7665 .08707 [cs.DB].
- 7666 Chen, Jinxiu, Donghong Ji, Chew Lim Tan, and Zhengyu Niu (July 2006). “Relation Extraction Using Label  
7667 Propagation Based Semi-Supervised Learning”. In: *Proceedings of the 21st International Conference on*  
7668



- 7669 *Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.  
7670 Sydney, Australia: Association for Computational Linguistics, pp. 129–136. DOI: 10.3115/1220175.1220  
7671 192. URL: <https://aclanthology.org/P06-1017>.
- 7672 Chevalier, Gil (1990). “Frontispice de la Bibliothèque Oucuienne”.
- 7673 Chinchor, Nancy A. (1998). “Overview of MUC-7”. In: *Seventh Message Understanding Conference (MUC-7):*  
7674 *Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. URL: <https://aclanthol>  
7675 [ogy.org/M98-1001](https://aclanthology.org/M98-1001).
- 7676 Cho, Kyunghyun, Bart van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Sch-  
7677 wenk, and Yoshua Bengio (Oct. 2014). “Learning Phrase Representations using RNN Encoder–Decoder for  
7678 Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural*  
7679 *Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734.  
7680 DOI: 10.3115/v1/D14-1179. URL: <https://www.aclweb.org/anthology/D14-1179>.
- 7681 Cohen, Amir DN, Shachar Rosenman, and Yoav Goldberg (2021). “Relation Classification as Two-way Span-  
7682 Prediction”. Under review for ACL 2022. arXiv: 2010.04829 [cs.CL]. URL: <https://arxiv.org/abs/201>  
7683 [0.04829](https://arxiv.org/abs/2010.04829).
- 7684 Collobert, Ronan and Jason Weston (2008). “A unified architecture for natural language processing: deep  
7685 neural networks with multitask learning”. In: ed. by Andrew McCallum and Sam Roweis, pp. 160–167.  
7686 DOI: 10.1145/1390156.1390177. URL: <https://dl.acm.org/doi/pdf/10.1145/1390156.1390177>.
- 7687 Conard, Louis (1926). “Lettre du 16 mai 1843 à sa sœur”. In: *Correspondance de Gustave Flaubert*. Vol. 1,  
7688 pp. 139–140.
- 7689 Conneau, Alexis and Guillaume Lample (2019). “Cross-lingual Language Model Pretraining”. In: *Advances in*  
7690 *Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc,  
7691 E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper>  
7692 [/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf](https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf).
- 7693 Cortes, Corinna and Vladimir Vapnik (1995). “Support-vector networks”. In: *Machine learning* 20.3, pp. 273–  
7694 297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.
- 7695 Craven, Mark and Johan Kumlien (1999). “Constructing biological knowledge bases by extracting information  
7696 from text sources”. In: *Proceedings of the Seventh International Conference on Intelligent Systems for*  
7697 *Molecular Biology*. Vol. 1999, pp. 77–86. URL: <https://www.aaai.org/Papers/ISMB/1999/ISMB99-010>  
7698 [.pdf](https://www.aaai.org/Papers/ISMB/1999/ISMB99-010.pdf).
- 7699 Culotta, Aron and Jeffrey Sorensen (July 2004). “Dependency Tree Kernels for Relation Extraction”. In:  
7700 *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona,  
7701 Spain, pp. 423–429. DOI: 10.3115/1218955.1219009. URL: <https://aclanthology.org/P04-1054>.
- 7702 Cuturi, Marco (2013). “Sinkhorn Distances: Lightspeed Computation of Optimal Transport”. In: *Advances in*  
7703 *Neural Information Processing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani,  
7704 and K. Q. Weinberger. Vol. 26. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper>  
7705 [/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf](https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf).
- 7706 Cybenko, George (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of*  
7707 *control, signals and systems* 2.4, pp. 303–314.
- 7708 Dalton, Jeffrey, Laura Dietz, and James Allan (2014). “Entity Query Feature Expansion Using Knowledge  
7709 Base Links”. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development*  
7710 *in Information Retrieval*. SIGIR ’14. Gold Coast, Queensland, Australia: ACM, pp. 365–374. ISBN: 978-1-  
7711 4503-2257-7. DOI: 10.1145/2600428.2609628. URL: <http://doi.acm.org/10.1145/2600428.2609628>.
- 7712 Darroch, John Newton and D. Ratcliff (1972). “Generalized Iterative Scaling for Log-Linear Models”. In: *The*  
7713 *Annals of Mathematical Statistics* 43.5, pp. 1470–1480. ISSN: 00034851. URL: <http://www.jstor.org/sta>  
7714 [ble/2240069](http://www.jstor.org/stable/2240069).
- 7715 Defays, Daniel (1977). “An efficient algorithm for a complete link method”. In: *The Computer Journal* 20.4,  
7716 pp. 364–366.
- 7717 Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). “Convolutional Neural Networks on  
7718 Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems*.  
7719 Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc. URL:  
7720 [https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.p](https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf)  
7721 [df](https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf).  
7722

- 7723 Saussure, Ferdinand de (1916). *Cours de linguistique générale*. French. Ed. by Albert Bally Charles et Seche-  
7724 haye. Payot.
- 7725 Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). “BERT: Pre-training of Deep  
7726 Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the*  
7727 *North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*  
7728 *Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics,  
7729 pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://www.aclweb.org/anthology/N19-1423>.
- 7730 Dietterich, Thomas G., Richard H. Lathrop, and Tomás Lozano-Pérez (1997). “Solving the multiple instance  
7731 problem with axis-parallel rectangles”. In: *Artificial Intelligence* 89.1, pp. 31–71. ISSN: 0004-3702. DOI:  
7732 [https://doi.org/10.1016/S0004-3702\(96\)00034-3](https://doi.org/10.1016/S0004-3702(96)00034-3). URL: [https://www.sciencedirect.com/science](https://www.sciencedirect.com/science/article/pii/S0004370296000343)  
7733 [/article/pii/S0004370296000343](https://www.sciencedirect.com/science/article/pii/S0004370296000343).
- 7734 Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and  
7735 Ralph M Weischedel (2004). “The automatic content extraction (ACE) program-tasks, data, and evalua-  
7736 tion.” In: 2.1, pp. 837–840. URL: [https://www ldc.upenn.edu/sites/www ldc.upenn.edu/files/lrec](https://www ldc.upenn.edu/sites/www ldc.upenn.edu/files/lrec 2004-ace-program.pdf)  
7737 [2004-ace-program.pdf](https://www ldc.upenn.edu/sites/www ldc.upenn.edu/files/lrec 2004-ace-program.pdf).
- 7738 Dostert, Leon E (1955). “The Georgetown–IBM experiment”. In: *Machine translation of languages*, pp. 124–  
7739 135.
- 7740 Downey, Doug, Oren Etzioni, and Stephen Soderland (2005). “A probabilistic model of redundancy in infor-  
7741 mation extraction”. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*,  
7742 pp. 1028–1033. URL: <https://www.ijcai.org/Proceedings/05/Papers/1390.pdf>.
- 7743 Dumais, Susan T, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman (1988).  
7744 “Using latent semantic analysis to improve access to textual information”. In: *Proceedings of the SIGCHI*  
7745 *conference on Human factors in computing systems*, pp. 281–285. DOI: 10.1145/57167.57214. URL:  
7746 <https://dl.acm.org/doi/pdf/10.1145/57167.57214>.
- 7747 Elsahar, Hady, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest,  
7748 and Elena Simperl (May 2018). “T-REX: A Large Scale Alignment of Natural Language with Knowledge  
7749 Base Triples”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evalu-*  
7750 *ation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: [https://a](https://aclanthology.org/L18-1544)  
7751 [clanthology.org/L18-1544](https://aclanthology.org/L18-1544).
- 7752 Fraser, Chris (2007). “Language and Ontology in Early Chinese Thought”. In: *Philosophy East and West* 57.4,  
7753 pp. 420–456. ISSN: 00318221, 15291898. URL: <http://www.jstor.org/stable/20109423>.
- 7754 Freund, Yoav and Robert E. Schapire (1999). “Large margin classification using the perceptron algorithm”.  
7755 In: *Machine learning* 37.3, pp. 277–296. ISSN: 1573-0565. DOI: 10.1023/A:1007662407062.
- 7756 Fu, Tsu-Jui, Peng-Hsuan Li, and Wei-Yun Ma (July 2019). “GraphRel: Modeling Text as Relational Graphs  
7757 for Joint Entity and Relation Extraction”. In: *Proceedings of the 57th Annual Meeting of the Association*  
7758 *for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1409–1418.  
7759 DOI: 10.18653/v1/P19-1136. URL: <https://aclanthology.org/P19-1136>.
- 7760 Gage, Philip (1994). “A new algorithm for data compression”. In: *C Users Journal* 12.2, pp. 23–38.
- 7761 Gao, Tianyu, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou (Nov. 2019). “FewRel 2.0:  
7762 Towards More Challenging Few-Shot Relation Classification”. In: *Proceedings of the 2019 Conference on*  
7763 *Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*  
7764 *Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics,  
7765 pp. 6250–6255. DOI: 10.18653/v1/D19-1649. URL: <https://aclanthology.org/D19-1649>.
- 7766 Gene Ontology Consortium (Jan. 2004). “The Gene Ontology (GO) database and informatics resource”. In:  
7767 *Nucleic Acids Research* 32, pp. D258–D261. ISSN: 0305-1048. DOI: 10.1093/nar/gkh036. URL: [https://a](https://academic.oup.com/nar/article-pdf/32/suppl%5C_1/D258/7621365/gkh036.pdf)  
7768 [cademic.oup.com/nar/article-pdf/32/suppl%5C\\_1/D258/7621365/gkh036.pdf](https://academic.oup.com/nar/article-pdf/32/suppl%5C_1/D258/7621365/gkh036.pdf).
- 7769 Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). “Deep Sparse Rectifier Neural Networks”. In:  
7770 *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (Apr. 11–  
7771 13, 2011). Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine  
7772 Learning Research. Fort Lauderdale, FL, USA, pp. 315–323. URL: [http://proceedings.mlr.press/v15/g](http://proceedings.mlr.press/v15/glorot11a.html)  
7773 [lorot11a.html](http://proceedings.mlr.press/v15/glorot11a.html).
- 7774 Google (2016). *Freebase Data Dumps*. URL: <https://developers.google.com/freebase/data>.  
7775  
7776

- 7777 Gracia, Jorge and Lloyd Newton (2016). “Medieval Theories of the Categories”. In: *The Stanford Encyclopedia*  
7778 *of Philosophy*. Ed. by Edward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University. URL:  
7779 <https://plato.stanford.edu/archives/win2016/entries/medieval-categories/>.
- 7780 Greff, Klaus, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber (2017). “LSTM:  
7781 A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10, pp. 2222–  
7782 2232. DOI: 10.1109/TNNLS.2016.2582924.
- 7783 Greff, Klaus, Sjoerd van Steenkiste, and Jürgen Schmidhuber (2020). “On the Binding Problem in Artificial  
7784 Neural Networks”. arXiv: 2012.05208 [cs.NE].
- 7785 Gumbel, Emil Julius (1954). *Statistical Theory of Extreme Values and Some Practical Applications. A Series*  
7786 *of Lectures*. US Government Printing Office. URL: [https://ntrl.ntis.gov/NTRL/dashboard/searchRes](https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/PB175818.xhtml)  
7787 [ults/titleDetail/PB175818.xhtml](https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/PB175818.xhtml).
- 7788 Gutmann, Michael and Aapo Hyvärinen (2010). “Noise-contrastive estimation: A new estimation principle  
7789 for unnormalized statistical models”. In: *Proceedings of the Thirteenth International Conference on Arti-*  
7790 *ficial Intelligence and Statistics* (May 13–15, 2010). Ed. by Yee Whye Teh and Mike Titterton. Vol. 9.  
7791 Proceedings of Machine Learning Research. JMLR Workshop and Conference Proceedings. Chia Laguna  
7792 Resort, Sardinia, Italy, pp. 297–304. URL: <http://proceedings.mlr.press/v9/gutmann10a.html>.
- 7793 Hamilton, Will, Zhitao Ying, and Jure Leskovec (2017). “Inductive Representation Learning on Large Graphs”.  
7794 In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H.  
7795 Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. URL: [https://pr](https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf)  
7796 [oceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf](https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf).
- 7797 Han, Xu, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun (Oct. 2018). “FewRel:  
7798 A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation”.  
7799 In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels,  
7800 Belgium: Association for Computational Linguistics, pp. 4803–4809. DOI: 10.18653/v1/D18-1514. URL:  
7801 <https://aclanthology.org/D18-1514>.
- 7802 Hansen, Chad D. (1983). *Language and logic in ancient China*. University of Michigan Press.
- 7803 Harbsmeier, Christoph (1989). “Marginalia sino-logica”. In: *Understanding the Chinese mind*, pp. 125–166.
- 7804 Harris, Zellig S. (1954). “Distributional Structure”. In: *WORD* 10.2–3, pp. 146–162. DOI: 10.1080/00437956.1  
7805 954.11659520.
- 7806 Hasegawa, Takaaki, Satoshi Sekine, and Ralph Grishman (July 2004). “Discovering Relations among Named  
7807 Entities from Large Corpora”. In: *Proceedings of the 42nd Annual Meeting of the Association for Com-*  
7808 *putational Linguistics (ACL-04)*. Barcelona, Spain, pp. 415–422. DOI: 10.3115/1218955.1219008. URL:  
7809 <https://aclanthology.org/P04-1053>.
- 7810 Hearst, Marti A. (1992). “Automatic Acquisition of Hyponyms from Large Text Corpora”. In: *COLING 1992*  
7811 *Volume 2: The 14th International Conference on Computational Linguistics*. URL: [https://aclantholog](https://aclanthology.org/C92-2082)  
7812 [y.org/C92-2082](https://aclanthology.org/C92-2082).
- 7813 Hendrickx, Iris, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó,  
7814 Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz (July 2010). “SemEval-2010 Task 8: Multi-  
7815 Way Classification of Semantic Relations between Pairs of Nominals”. In: *Proceedings of the 5th Interna-*  
7816 *tional Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics,  
7817 pp. 33–38. URL: <https://aclanthology.org/S10-1006>.
- 7818 Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mo-  
7819 hamed, and Alexander Lerchner (2017). “ $\beta$ -VAE: Learning Basic Visual Concepts with a Constrained Vari-  
7820 ational Framework”. In: *International Conference on Learning Representations*. URL: [https://openrevie](https://openreview.net/forum?id=Sy2fzU9gl)  
7821 [w.net/forum?id=Sy2fzU9gl](https://openreview.net/forum?id=Sy2fzU9gl).
- 7822 Hinton, Geoffrey E (1986). “Learning distributed representations of concepts”. In: *Proceedings of the eighth*  
7823 *annual conference of the cognitive science society*. Vol. 1. Amherst, MA, USA, p. 12. URL: [https://www.cs](https://www.cs.toronto.edu/~hinton/absps/families.pdf)  
7824 [.toronto.edu/~hinton/absps/families.pdf](https://www.cs.toronto.edu/~hinton/absps/families.pdf).
- 7825 Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh (July 2006). “A Fast Learning Algorithm for Deep  
7826 Belief Nets”. In: *Neural Computation* 18.7, pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18  
7827 .7.1527. URL: <https://direct.mit.edu/neco/article/18/7/1527/7065>.
- 7828
- 7829
- 7830

- 7831 Hochreiter, Sepp (Apr. 1998). “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and  
7832 Problem Solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6,  
7833 pp. 107–116. DOI: 10.1142/S0218488598000094.
- 7834 Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Computation*  
7835 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: [https://direct.mit.ed  
7836 u/neco/article/9/8/1735/6109](https://direct.mit.edu/neco/article/9/8/1735/6109).
- 7837 Hoffmann, Raphael, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel Weld (June 2011). “Knowledge-  
7838 Based Weak Supervision for Information Extraction of Overlapping Relations”. In: *Proceedings of the 49th*  
7839 *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland,  
7840 Oregon, USA: Association for Computational Linguistics, pp. 541–550. URL: [https://aclanthology.org  
7841 /P11-1055](https://aclanthology.org/P11-1055).
- 7842 Hu, Xuming, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu (Nov. 2020). “SelfORE: Self-supervised  
7843 Relational Feature Learning for Open Relation Extraction”. In: *Proceedings of the 2020 Conference on*  
7844 *Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Lin-  
7845 guistics, pp. 3673–3682. DOI: 10.18653/v1/2020.emnlp-main.299. URL: [https://aclanthology.org/20  
7846 20.emnlp-main.299](https://aclanthology.org/2020.emnlp-main.299).
- 7847 Hu, Ziniu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun (2020). “Heterogeneous Graph Transformer”. In:  
7848 *Proceedings of The Web Conference 2020*. New York, NY, USA: Association for Computing Machinery,  
7849 pp. 2704–2710. ISBN: 9781450370233. DOI: 10.1145/3366423.3380027. URL: [https://dl.acm.org/doi/p  
7850 df/10.1145/3366423.3380027](https://dl.acm.org/doi/pdf/10.1145/3366423.3380027).
- 7851 Hubert, Lawrence and Phipps Arabie (Dec. 1985). “Comparing partitions”. In: *Journal of classification* 2.1,  
7852 pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075. URL: [https://link.springer.com/content/p  
7853 df/10.1007/BF01908075.pdf](https://link.springer.com/content/pdf/10.1007/BF01908075.pdf).
- 7854 Immerman, Neil and Eric Lander (1990). “Describing Graphs: A First-Order Approach to Graph Canoniza-  
7855 tion”. In: *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth*  
7856 *Birthday, July 5, 1988*. Ed. by Alan L. Selman. New York, NY, USA: Springer New York, pp. 59–81. ISBN:  
7857 978-1-4612-4478-3. DOI: 10.1007/978-1-4612-4478-3\_5. URL: [https://www.cs.yale.edu/publication  
7858 s/techreports/tr605.pdf](https://www.cs.yale.edu/publications/techreports/tr605.pdf).
- 7859 Jang, Eric, Shixiang Gu, and Ben Poole (2016). “Categorical reparameterization with gumbel-softmax”. In:  
7860 *International Conference on Learning Representations*. URL: [https://openreview.net/forum?id=rkE3y  
7861 85ee](https://openreview.net/forum?id=rkE3y85ee).
- 7862 Jarry, Alfred (1911). *Gestes et opinions du docteur Faustroll*.
- 7863 Jiang, Tianwen, Sendong Zhao, Jing Liu, Jin-Ge Yao, Ming Liu, Bing Qin, Ting Liu, and Chin-Yew Lin (2019).  
7864 “Towards Time-Aware Distant Supervision for Relation Extraction”. arXiv: 1903.03289 [cs.CL].
- 7865 Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu (2016). “Exploring the Limits  
7866 of Language Modeling”. arXiv: 1602.02410 [cs.CL].
- 7867 Kambhatla, Nanda (July 2004). “Combining Lexical, Syntactic, and Semantic Features with Maximum En-  
7868 tropy Models for Information Extraction”. In: *Proceedings of the ACL Interactive Poster and Demonstration*  
7869 *Sessions*. Barcelona, Spain: Association for Computational Linguistics, pp. 178–181. URL: [https://aclan  
7870 thology.org/P04-3022](https://aclanthology.org/P04-3022).
- 7871 Kim, Yoon (Oct. 2014). “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the*  
7872 *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association  
7873 for Computational Linguistics, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: [https://www.aclweb.o  
7874 rg/anthology/D14-1181](https://www.aclweb.org/anthology/D14-1181).
- 7875 Kingma, Diederik P. and Max Welling (2014). “Auto-Encoding Variational Bayes”. In: *2nd International*  
7876 *Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference*  
7877 *Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: <http://arxiv.org/abs/1312.6114>.
- 7878 Kipf, Thomas N and Max Welling (2017). “Semi-Supervised Classification with Graph Convolutional Net-  
7879 works”. In: *International Conference on Learning Representations*. URL: [https://openreview.net/foru  
7880 m?id=SJU4ayYgl](https://openreview.net/forum?id=SJU4ayYgl).
- 7881 Klein, Dan and Christopher Manning (July 2003). “Accurate Unlexicalized Parsing”. In: *Proceedings of the*  
7882 *41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for  
7883  
7884

- 7885 Computational Linguistics, pp. 423–430. DOI: 10.3115/1075096.1075150. URL: [https://aclanthology](https://aclanthology.org/P03-1054)  
7886 [.org/P03-1054](https://aclanthology.org/P03-1054).
- 7887 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “ImageNet Classification with Deep Con-  
7888 volutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira,  
7889 C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc. URL: [https://proce-](https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)  
7890 [edings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf).
- 7891 LeCun, Yann and Ishan Misra (Mar. 4, 2021). *Self-supervised learning: The dark matter of intelligence*. URL:  
7892 <https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence>  
7893 (visited on 11/08/2021).
- 7894 Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang  
7895 (Sept. 2019). “BioBERT: a pre-trained biomedical language representation model for biomedical text min-  
7896 ing”. In: *Bioinformatics* 36.4, pp. 1234–1240. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682.  
7897 URL: <https://academic.oup.com/bioinformatics/article-pdf/36/4/1234/32527770/btz682.pdf>.
- 7898 Leshno, Moshe, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken (1993). “Multilayer feedforward networks  
7899 with a nonpolynomial activation function can approximate any function”. In: *Neural networks* 6.6, pp. 861–  
7900 867.
- 7901 Levy, Omer and Yoav Goldberg (2014). “Neural Word Embedding as Implicit Matrix Factorization”. In:  
7902 *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N.  
7903 Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc. URL: [https://proceedings.neurips](https://proceedings.neurips.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf)  
7904 [.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf](https://proceedings.neurips.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf).
- 7905 Lin, Dekang and Patrick Pantel (2001). “DIRT – Discovery of Inference Rules from Text”. In: *Proceedings of the*  
7906 *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco,  
7907 California: Association for Computing Machinery, pp. 323–328. ISBN: 158113391X. DOI: 10.1145/502512  
7908 .502559. URL: <http://www.patrickpantel.com/download/papers/2001/kdd01-1.pdf>.
- 7909 Lin, Yankai, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu (2015). “Learning Entity and Relation  
7910 Embeddings for Knowledge Graph Completion”. In: *Proceedings of the Twenty-Ninth AAAI Conference on*  
7911 *Artificial Intelligence*. Austin, Texas: AAAI Press, pp. 2181–2187. ISBN: 0262511290.
- 7912 Lin, Yankai, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun (Aug. 2016). “Neural Relation Extrac-  
7913 tion with Selective Attention over Instances”. In: *Proceedings of the 54th Annual Meeting of the Association*  
7914 *for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational  
7915 Linguistics, pp. 2124–2133. DOI: 10.18653/v1/P16-1200. URL: <https://aclanthology.org/P16-1200>.
- 7916 Maas, Andrew L, Awni Y Hannun, Andrew Y Ng, et al. (2013). “Rectifier nonlinearities improve neural network  
7917 acoustic models”. In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*.  
7918 Vol. 30. 1, p. 3. URL: [https://ai.stanford.edu/~amaas/papers/relu\\_hybrid\\_icml2013\\_final.pdf](https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf).
- 7919 Marcheggiani, Diego and Ivan Titov (2016). “Discrete-State Variational Autoencoders for Joint Discovery and  
7920 Factorization of Relations”. In: *Transactions of the Association for Computational Linguistics* 4, pp. 231–  
7921 244. DOI: 10.1162/tacl\_a\_00095. URL: <https://aclanthology.org/Q16-1017>.
- 7922 Marque-Pucheu, Christiane (2008). “La couleur des prépositions à et de”. In: vol. 157. Paris, France: Armand  
7923 Colin, pp. 74–105. DOI: 10.3917/lf.157.0074. URL: [https://www.cairn.info/load\\_pdf.php](https://www.cairn.info/load_pdf.php?ID_ARTICLE=LF_157_0074)  
7924 [?ID\\_ARTICLE=LF\\_157\\_0074](https://www.cairn.info/load_pdf.php?ID_ARTICLE=LF_157_0074).
- 7925 Mathon, Rudolf (1979). “A note on the graph isomorphism counting problem”. In: *Information Processing*  
7926 *Letters* 8.3, pp. 131–136.
- 7927 McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher (2017). “Learned in Translation: Con-  
7928 textualized Word Vectors”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V.  
7929 Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates,  
7930 Inc. URL: [https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12](https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12-Paper.pdf)  
7931 [-Paper.pdf](https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12-Paper.pdf).
- 7932 McCarthy, John (1959). “Programs with common sense”. In: URL: [http://www-formal.stanford.edu/jmc](http://www-formal.stanford.edu/jmc/mcc59/mcc59.html)  
7933 [/mcc59/mcc59.html](http://www-formal.stanford.edu/jmc/mcc59/mcc59.html).
- 7934 McDonald, Ryan, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White (June 2005).  
7935 “Simple Algorithms for Complex Relation Extraction with Applications to Biomedical IE”. In: *Proceed-*  
7936 *ings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. Ann Arbor,  
7937  
7938

- 7939 Michigan: Association for Computational Linguistics, pp. 491–498. DOI: 10.3115/1219840.1219901. URL:  
7940 <https://aclanthology.org/P05-1061>.
- 7941 Mendes, Pablo N., Max Jakob, Andrés García-Silva, and Christian Bizer (2011). “DBpedia Spotlight: Shedding  
7942 Light on the Web of Documents”. In: *Proceedings of the 7th International Conference on Semantic Systems.*  
7943 *I-Semantics ’11*. Graz, Austria: Association for Computing Machinery, pp. 1–8. ISBN: 9781450306218. DOI:  
7944 10.1145/2063518.2063519. URL: <https://dl.acm.org/doi/pdf/10.1145/2063518.2063519>.
- 7945 Mesquita, Filipe, Matteo Cannavicchio, Jordan Schmeidek, Paramita Mirza, and Denilson Barbosa (Nov. 2019).  
7946 “KnowledgeNet: A Benchmark Dataset for Knowledge Base Population”. In: *Proceedings of the 2019 Con-*  
7947 *ference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*  
7948 *on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Lin-  
7949 guistics, pp. 749–758. DOI: 10.18653/v1/D19-1069. URL: <https://aclanthology.org/D19-1069>.
- 7950 Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). “Efficient Estimation of Word Repre-  
7951 sentations in Vector Space”. arXiv: 1301.3781 [cs.CL].
- 7952 Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013b). “Distributed Represen-  
7953 tations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing*  
7954 *Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26.  
7955 Curran Associates, Inc. URL: [https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039](https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf)  
7956 [965f3c4923ce901b-Paper.pdf](https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf).
- 7957 Miller, George A. (Nov. 1995). “WordNet: A Lexical Database for English”. In: *Communications of the ACM*  
7958 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748.
- 7959 Miller, Scott, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weische-  
7960 del, and The Annotation Group (1998). “BBN: Description of the SIFT System as Used for MUC-7”. In:  
7961 *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Vir-*  
7962 *ginia, April 29 – May 1, 1998*. URL: <https://aclanthology.org/M98-1009>.
- 7963 Mintz, Mike, Steven Bills, Rion Snow, and Daniel Jurafsky (Aug. 2009). “Distant supervision for relation  
7964 extraction without labeled data”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of*  
7965 *the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec,  
7966 Singapore: Association for Computational Linguistics, pp. 1003–1011. URL: [https://aclanthology.org](https://aclanthology.org/P09-1113)  
7967 [/P09-1113](https://aclanthology.org/P09-1113).
- 7968 Mnih, Andriy and Yee Whye Teh (2012). “A fast and simple algorithm for training neural probabilistic  
7969 language models”. In: *Proceedings of the 29th International Conference on Machine Learning*, p. 58. URL:  
7970 <http://icml.cc/2012/papers/855.pdf>.
- 7971 Montariol, Syrielle, Étienne Simon, Arij Riabi, and Djamé Seddah (May 2022). “Fine-tuning and Sampling  
7972 Strategies for Multimodal Role Labeling of Entities under Class Imbalance”. In: *Proceedings of the Workshop*  
7973 *on Combating Online Hostile Posts in Regional Languages during Emergency Situations*. Dublin, Ireland:  
7974 Association for Computational Linguistics, pp. 55–65. URL: [https://aclanthology.org/2022.constrai](https://aclanthology.org/2022.constraint-1.7)  
7975 [nt-1.7](https://aclanthology.org/2022.constraint-1.7).
- 7976 Morgan, Augustus De (1864). “On the Syllogism, No. III, and on Logic in general”. In: *Transactions of the*  
7977 *Cambridge Philosophical Society* 10, pp. 173–230.
- 7978 Morris, Christopher, Nils M. Kriege, Kristian Kersting, and Petra Mutzel (2016). “Faster Kernels for Graphs  
7979 with Continuous Attributes via Hashing”. In: *2016 IEEE 16th International Conference on Data Mining*  
7980 *(ICDM)*. IEEE, pp. 1095–1100. DOI: 10.1109/ICDM.2016.0142.
- 7981 Morris, Christopher, Gaurav Rattan, and Petra Mutzel (2020). “Weisfeiler and Leman go sparse: Towards  
7982 scalable higher-order graph embeddings”. In: *Advances in Neural Information Processing Systems*. Ed.  
7983 by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc.,  
7984 pp. 21824–21840. URL: [https://proceedings.neurips.cc/paper/2020/file/f81dee42585b3814de199](https://proceedings.neurips.cc/paper/2020/file/f81dee42585b3814de199b2e88757f5c-Paper.pdf)  
7985 [b2e88757f5c-Paper.pdf](https://proceedings.neurips.cc/paper/2020/file/f81dee42585b3814de199b2e88757f5c-Paper.pdf).
- 7986 Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel (June 2011). “A Three-Way Model for Collective  
7987 Learning on Multi-Relational Data”. In: *Proceedings of the 28th International Conference on Machine*  
7988 *Learning (ICML-11)*. Ed. by Lise Getoor and Tobias Scheffer. Bellevue, WA, USA: ACM, pp. 809–816. ISBN:  
7989 978-1-4503-0619-5. URL: [https://icml.cc/2011/papers/438\\_icmlpaper.pdf](https://icml.cc/2011/papers/438_icmlpaper.pdf).
- 7990 Norvig, Peter (2011). *On Chomsky and the Two Cultures of Statistical Learning*. URL: [https://norvig.com](https://norvig.com/chomsky.html)  
7991 [/chomsky.html](https://norvig.com/chomsky.html).
- 7992

- 7993 Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). “GloVe: Global Vectors for Word  
7994 Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Pro-*  
7995 *cessing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115  
7996 /v1/D14-1162. URL: <https://www.aclweb.org/anthology/D14-1162>.
- 7997 Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). “DeepWalk: Online Learning of Social Representa-  
7998 tions”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data*  
7999 *Mining*. New York, NY, USA: Association for Computing Machinery, pp. 701–710. ISBN: 9781450329569.  
8000 DOI: 10.1145/2623330.2623732. URL: <https://dl.acm.org/doi/pdf/10.1145/2623330.2623732>.
- 8001 Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke  
8002 Zettlemoyer (June 2018). “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Con-*  
8003 *ference of the North American Chapter of the Association for Computational Linguistics: Human Language*  
8004 *Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics,  
8005 pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: <https://www.aclweb.org/anthology/N18-1202>.
- 8006 Poincaré, Henri (1908). *Thermodynamique*. Gauthier-Villars.
- 8007 Qian, Yujie, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay (June 2019). “GraphIE: A Graph-  
8008 Based Framework for Information Extraction”. In: *Proceedings of the 2019 Conference of the North Amer-*  
8009 *ican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*  
8010 *(Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 751–761.  
8011 DOI: 10.18653/v1/N19-1082. URL: <https://aclanthology.org/N19-1082>.
- 8012 Qu, Meng, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang (July 2020). “Few-shot Relation Extraction  
8013 via Bayesian Meta-learning on Relation Graphs”. In: *Proceedings of the 37th International Conference on*  
8014 *Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning  
8015 Research. PMLR, pp. 7867–7876. URL: <https://proceedings.mlr.press/v119/qu20a.html>.
- 8016 Quine, Willard Van Orman (1951). “Main Trends in Recent Philosophy: Two Dogmas of Empiricism”. In: *The*  
8017 *Philosophical Review* 60.1, pp. 20–43. ISSN: 00318108, 15581470. URL: [http://www.jstor.org/stable/2](http://www.jstor.org/stable/2181906)  
8018 181906.
- 8019 — (2004). *Du point de vue logique : neuf essais logico-philosophiques*. Trans. by Sandra Laugier. Vrin.
- 8020 Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). “Improving Language Under-  
8021 standing by Generative Pre-Training”.
- 8022 Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou,  
8023 Wei Li, and Peter J. Liu (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text  
8024 Transformer”. In: *Journal of Machine Learning Research* 21.140, pp. 1–67. URL: [http://jmlr.org/paper](http://jmlr.org/papers/v21/20-074.html)  
8025 [s/v21/20-074.html](http://jmlr.org/papers/v21/20-074.html).
- 8026 Rand, William M. (1971). “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the*  
8027 *American Statistical Association* 66.336, pp. 846–850. DOI: 10.1080/01621459.1971.10482356.
- 8028 Redouté, Pierre-Joseph (1802). “Paris Quadrifolia”. In: *Les Liliacées*. URL: [https://commons.wikimedia.or](https://commons.wikimedia.org/wiki/File:Paris_quadrifolia_in_Les_liliacees.jpg)  
8029 [g/wiki/File:Paris\\_quadrifolia\\_in\\_Les\\_liliacees.jpg](https://commons.wikimedia.org/wiki/File:Paris_quadrifolia_in_Les_liliacees.jpg). Via Wikimedia Commons.
- 8030 Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme (2009). “BPR: Bayesian  
8031 Personalized Ranking from Implicit Feedback”. In: *Proceedings of the Twenty-Fifth Conference on Uncer-*  
8032 *tainty in Artificial Intelligence*. Montreal, Quebec, Canada: AUAI Press, pp. 452–461. ISBN: 9780974903958.  
8033 DOI: 10.5555/1795114.1795167. URL: <https://dl.acm.org/doi/pdf/10.5555/1795114.1795167>.
- 8034 Riedel, Sebastian, Limin Yao, Andrew McCallum, and Benjamin Marlin (June 2013). “Relation Extraction  
8035 with Matrix Factorization and Universal Schemas”. In: *Proceedings of the 2013 Conference of the North*  
8036 *American Chapter of the Association for Computational Linguistics: Human Language Technologies*. At-  
8037 lanta, Georgia: Association for Computational Linguistics, pp. 74–84. URL: [https://aclanthology.org](https://aclanthology.org/N13-1008)  
8038 [/N13-1008](https://aclanthology.org/N13-1008).
- 8039 Roberts, Ben and Dirk P Kroese (2007). “Estimating the Number of  $s-t$  Paths in a Graph.” In: *Journal of*  
8040 *Graph Algorithms and Applications* 11.1, pp. 195–214.
- 8041 Rosenberg, Andrew and Julia Hirschberg (June 2007). “V-Measure: A Conditional Entropy-Based External  
8042 Cluster Evaluation Measure”. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Nat-*  
8043 *ural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech  
8044 Republic: Association for Computational Linguistics, pp. 410–420. URL: [https://aclanthology.org/D0](https://aclanthology.org/D07-1043)  
8045 [7-1043](https://aclanthology.org/D07-1043).
- 8046

- 8047 Sager, Naomi (1972). “Syntactic Formatting of Science Information”. In: *Proceedings of the December 5-7, 1972,*  
8048 *Fall Joint Computer Conference, Part II*. Anaheim, California: Association for Computing Machinery,  
8049 pp. 791–800. ISBN: 9781450379137. DOI: 10.1145/1480083.1480101. URL: [https://dl.acm.org/doi/pdf/](https://dl.acm.org/doi/pdf/10.1145/1480083.1480101)  
8050 [f/10.1145/1480083.1480101](https://dl.acm.org/doi/pdf/10.1145/1480083.1480101).
- 8051 Sandhaus, Evan (2008). *The New York Times Annotated Corpus*. LDC2008T19. Philadelphia: Linguistic Data  
8052 Consortium. DOI: 10.35111/77ba-9x74. URL: <https://catalog.ldc.upenn.edu/LDC2008T19>.
- 8053 Schlichtkrull, Michael, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling  
8054 (2018). “Modeling Relational Data with Graph Convolutional Networks”. In: *The Semantic Web*. Ed. by  
8055 Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna  
8056 Tordai, and Mehwish Alam. Cham: Springer International Publishing, pp. 593–607. ISBN: 978-3-319-93417-  
8057 4. URL: <https://arxiv.org/pdf/1703.06103.pdf>.
- 8058 Shuman, David I, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst (2013). “The  
8059 emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and  
8060 other irregular domains”. In: *IEEE Signal Processing Magazine* 30.3, pp. 83–98. DOI: 10.1109/MSP.2012.2  
8061 235192. URL: <https://arxiv.org/pdf/1211.0053.pdf>.
- 8062 Simon, Étienne, Vincent Guigue, and Benjamin Piwowarski (July 2019). “Unsupervised Information Extrac-  
8063 tion: Regularizing Discriminative Approaches with Relation Distribution Losses”. In: *Proceedings of the*  
8064 *57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for  
8065 Computational Linguistics, pp. 1378–1387. DOI: 10.18653/v1/P19-1133. URL: [https://www.aclweb.org](https://www.aclweb.org/anthology/P19-1133)  
8066 [/anthology/P19-1133](https://www.aclweb.org/anthology/P19-1133).
- 8067 Soames, Scott (1997). “Skepticism about Meaning: Indeterminacy, Normativity, and the Rule-Following Para-  
8068 dox”. In: *Canadian Journal of Philosophy Supplementary Volume* 23, pp. 211–249. DOI: 10.1080/0045509  
8069 1.1997.10715967.
- 8070 Soares, Livio Baldini, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski (July 2019). “Matching the  
8071 Blanks: Distributional Similarity for Relation Learning”. In: *Proceedings of the 57th Annual Meeting of*  
8072 *the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics,  
8073 pp. 2895–2905. DOI: 10.18653/v1/P19-1279. URL: <https://aclanthology.org/P19-1279>.
- 8074 Socher, Richard, Danqi Chen, Christopher D Manning, and Andrew Ng (2013). “Reasoning With Neural Tensor  
8075 Networks for Knowledge Base Completion”. In: *Advances in Neural Information Processing Systems*. Ed. by  
8076 C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates,  
8077 Inc. URL: [https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80](https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf)  
8078 [-Paper.pdf](https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf).
- 8079 Socher, Richard, Brody Huval, Christopher D. Manning, and Andrew Y. Ng (July 2012). “Semantic Com-  
8080 positionality through Recursive Matrix-Vector Spaces”. In: *Proceedings of the 2012 Joint Conference on*  
8081 *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju  
8082 Island, Korea: Association for Computational Linguistics, pp. 1201–1211. URL: [https://aclanthology.o](https://aclanthology.org/D12-1110)  
8083 [rg/D12-1110](https://aclanthology.org/D12-1110).
- 8084 Sohn, Kihyuk, Honglak Lee, and Xinchun Yan (2015). “Learning Structured Output Representation using  
8085 Deep Conditional Generative Models”. In: *Advances in Neural Information Processing Systems*. Ed. by C.  
8086 Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc. URL: [https](https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf)  
8087 [://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf](https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf).
- 8088 Song, Linfeng, Yue Zhang, Zhiguo Wang, and Daniel Gildea (Oct. 2018). “N-ary Relation Extraction using  
8089 Graph-State LSTM”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*  
8090 *Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2226–2235. DOI: 10.18653  
8091 /v1/D18-1246. URL: <https://aclanthology.org/D18-1246>.
- 8092 Speaks, Jeff (2021). “Theories of Meaning”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N.  
8093 Zalta. Spring 2021. Metaphysics Research Lab, Stanford University. URL: [https://plato.stanford.edu](https://plato.stanford.edu/archives/spr2021/entries/meaning/)  
8094 [/archives/spr2021/entries/meaning/](https://plato.stanford.edu/archives/spr2021/entries/meaning/).
- 8095 Sperduti, A. and A. Starita (1997). “Supervised neural networks for the classification of structures”. In: *IEEE*  
8096 *Transactions on Neural Networks* 8.3, pp. 714–735. DOI: 10.1109/72.572108.
- 8097 Suárez, Jorge A (1983). *The mesoamerican indian languages*. Cambridge University Press.
- 8098 Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston, and Rob Fergus (2015). “End-To-End Memory Net-  
8099 works”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee,  
8100



- 8101 M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc. URL: [https://proceedings.neurips.cc](https://proceedings.neurips.cc/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf)  
8102 /paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.
- 8103 Surdeanu, Mihai, Julie Tibshirani, Ramesh Nallapati, and Christopher Manning (July 2012). “Multi-instance  
8104 Multi-label Learning for Relation Extraction”. In: *Proceedings of the 2012 Joint Conference on Empirical*  
8105 *Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island,  
8106 Korea: Association for Computational Linguistics, pp. 455–465. URL: [https://aclanthology.org/D12-1](https://aclanthology.org/D12-1042)  
8107 042.
- 8108 Sutskever, Ilya, James Martens, and Geoffrey Hinton (June 2011). “Generating Text with Recurrent Neural  
8109 Networks”. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. Ed.  
8110 by Lise Getoor and Tobias Scheffer. Bellevue, Washington, USA: Association for Computing Machinery,  
8111 pp. 1017–1024. ISBN: 978-1-4503-0619-5.
- 8112 Tang, Lei and Huan Liu (2009). “Relational Learning via Latent Social Dimensions”. In: *Proceedings of the*  
8113 *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris,  
8114 France: Association for Computing Machinery, pp. 817–826. ISBN: 9781605584959. DOI: 10.1145/1557019  
8115 .1557109. URL: <https://dl.acm.org/doi/pdf/10.1145/1557019.1557109>.
- 8116 Tenniel, John (1889). “Cheshire Cat details from the Tree Above Alice”. In: *The Nursery “Alice”*. URL: [http](http://commons.wikimedia.org/wiki/File:Tennel_Cheshire_proof.png)  
8117 [s://commons.wikimedia.org/wiki/File:Tennel\\_Cheshire\\_proof.png](http://commons.wikimedia.org/wiki/File:Tennel_Cheshire_proof.png). Via Wikimedia Commons.
- 8118 British Museum, the (100 BCE–100 CE). “Ariadne waking on the shore of Naxos”. URL: [https://www.british](https://www.britishmuseum.org/collection/image/254690001)  
8119 [museum.org/collection/image/254690001](https://www.britishmuseum.org/collection/image/254690001). Wall painting from Herculaneum, Asset number: 254690001,  
8120 Museum number: 1867,0508.1358.
- 8121 Togninalli, Matteo, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt (2019).  
8122 “Wasserstein Weisfeiler-Lehman Graph Kernels”. In: *Advances in Neural Information Processing Systems*.  
8123 Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran  
8124 Associates, Inc. URL: [https://proceedings.neurips.cc/paper/2019/file/73fed7fd472e502d8908794](https://proceedings.neurips.cc/paper/2019/file/73fed7fd472e502d8908794430511f4d-Paper.pdf)  
8125 430511f4d-Paper.pdf.
- 8126 Trisedya, Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang (July 2019). “Neural Relation  
8127 Extraction for Knowledge Base Enrichment”. In: *Proceedings of the 57th Annual Meeting of the Association*  
8128 *for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 229–240.  
8129 DOI: 10.18653/v1/P19-1023. URL: <https://aclanthology.org/P19-1023>.
- 8130 Turing, Alan Mathison (Oct. 1950). “Computing Machinery and Intelligence”. In: *Mind* LIX.236, pp. 433–460.  
8131 ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433. URL: [https://academic.oup.com/mind/article-pd](https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf)  
8132 [f/LIX/236/433/30123314/lix-236-433.pdf](https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf).
- 8133 Tyler, Andrea and Vyvyan Evans (2001). “Reconsidering prepositional polysemy networks: The case of over”.  
8134 In: *Language*, pp. 724–765.
- 8135 Ushio, Asahi, Jose Camacho-Collados, and Steven Schockaert (Nov. 2021). “Distilling Relation Embeddings  
8136 from Pretrained Language Models”. In: *Proceedings of the 2021 Conference on Empirical Methods in*  
8137 *Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational  
8138 Linguistics, pp. 9044–9062. DOI: 10.18653/v1/2021.emnlp-main.712. URL: [https://aclanthology.org](https://aclanthology.org/2021.emnlp-main.712)  
8139 /2021.emnlp-main.712.
- 8140 Valiant, Leslie G. (1979). “The Complexity of Enumeration and Reliability Problems”. In: *SIAM Journal on*  
8141 *Computing* 8.3, pp. 410–421. DOI: 10.1137/0208032.
- 8142 Oord, Aäron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalch-  
8143 brenner, Andrew Senior, and Koray Kavukcuoglu (2016). “WaveNet: A Generative Model for Raw Audio”.  
8144 arXiv: 1609.03499 [cs.SD].
- 8145 Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser,  
8146 and Illia Polosukhin (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing*  
8147 *Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R.  
8148 Garnett. Vol. 30. Curran Associates, Inc. URL: [https://proceedings.neurips.cc/paper/2017/file/3](https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)  
8149 [f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
- 8150 Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio  
8151 (2018). “Graph Attention Networks”. In: URL: <https://openreview.net/forum?id=rJXMpikCZ>.
- 8152 Vincent, Pascal, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol (2010).  
8153 “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local De-  
8154

- 8155 noising Criterion”. In: *Journal of Machine Learning Research* 11.110, pp. 3371–3408. URL: <http://jmlr.org/papers/v11/vincent10a.html>.
- 8156
- 8157 Vrandečić, Denny and Markus Krötzsch (Sept. 2014). “Wikidata: A Free Collaborative Knowledgebase”. In: *Communications of the ACM* 57.10, pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489. URL: <https://dl.acm.org/doi/pdf/10.1145/2629489>.
- 8158
- 8159
- 8160 Waibel, Alex, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang (1989). “Phoneme recognition using time-delay neural networks”. In: *IEEE transactions on acoustics, speech, and signal processing* 37.3, pp. 328–339.
- 8161
- 8162
- 8163 Wang, Xiao, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu (2019). “Heterogeneous Graph Attention Network”. In: *The World Wide Web Conference*. San Francisco, CA, USA: Association for Computing Machinery, pp. 2022–2032. ISBN: 9781450366748. DOI: 10.1145/3308558.3313562. URL: <https://dl.acm.org/doi/pdf/10.1145/3308558.3313562>.
- 8164
- 8165
- 8166
- 8167 Wang, Zhen, Jianwen Zhang, Jianlin Feng, and Zheng Chen (2014). “Knowledge Graph Embedding by Translating on Hyperplanes”. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI’14. Québec City, Québec, Canada: AAAI Press, pp. 1112–1119.
- 8168
- 8169
- 8170 Watterson, Bill (May 17, 1992). *Calvin and Hobbes*.
- 8171 Weisfeiler, Boris and Andrej Leman (1968). “The reduction of a graph to canonical form and the algebra which appears therein”. In: *NTI, Series* 2.9, pp. 12–16. URL: [https://www.iti.zcu.cz/wl2018/pdf/wl\\_paper\\_translation.pdf](https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf).
- 8172
- 8173
- 8174 Weston, Jason, Sumit Chopra, and Antoine Bordes (2015). “Memory Networks”. In: *3rd International Conference on Learning Representations (ICLR), Conference Track Proceedings* (May 7–9, 2015). Ed. by Yoshua Bengio and Yann LeCun. San Diego, CA, USA. URL: <http://arxiv.org/abs/1410.3916>.
- 8175
- 8176
- 8177 Xie, Junyuan, Ross Girshick, and Ali Farhadi (June 2016). “Unsupervised Deep Embedding for Clustering Analysis”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 478–487. URL: <https://proceedings.mlr.press/v48/xieb16.html>.
- 8178
- 8179
- 8180 Yamaguchi, Kouichi, Kenji Sakamoto, and Toshio Akabane (Nov. 1990). “A neural network for speaker-independent isolated word recognition”. In: *First International Conference on Spoken Language Processing*. Kobe, Japan, pp. 1077–1080. URL: [https://www.isca-speech.org/archive/icslp\\_1990/i90\\_1077.html](https://www.isca-speech.org/archive/icslp_1990/i90_1077.html).
- 8181
- 8182
- 8183
- 8184
- 8185 Yang, Zhilin, William Cohen, and Ruslan Salakhudinov (June 2016). “Revisiting Semi-Supervised Learning with Graph Embeddings”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, pp. 40–48. URL: <https://proceedings.mlr.press/v48/yanga16.html>.
- 8186
- 8187
- 8188
- 8189 Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf>.
- 8190
- 8191
- 8192
- 8193
- 8194 Yao, Limin, Aria Haghighi, Sebastian Riedel, and Andrew McCallum (July 2011). “Structured Relation Discovery using Generative Models”. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK: Association for Computational Linguistics, pp. 1456–1466. URL: <https://aclanthology.org/D11-1135>.
- 8195
- 8196
- 8197
- 8198 Yao, Limin, Sebastian Riedel, and Andrew McCallum (July 2012). “Unsupervised Relation Discovery with Sense Disambiguation”. In: *Jeju Island, Korea: Association for Computational Linguistics*, pp. 712–720. URL: <https://aclanthology.org/P12-1075>.
- 8199
- 8200
- 8201 Yates, Alexander, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland (Apr. 2007). “TextRunner: Open Information Extraction on the Web”. In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. Rochester, NY, USA: Association for Computational Linguistics, pp. 25–26. URL: <https://aclanthology.org/N07-4013>.
- 8202
- 8203
- 8204
- 8205
- 8206 Yates, Alexander and Oren Etzioni (Apr. 2007). “Unsupervised Resolution of Objects and Relations on the Web”. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the*
- 8207
- 8208

- 8209 Association for Computational Linguistics; *Proceedings of the Main Conference*. Rochester, New York:  
8210 Association for Computational Linguistics, pp. 121–130. URL: <https://aclanthology.org/N07-1016>.
- 8211 Yih, Wen-tau, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao (2015). “Semantic Parsing via Staged Query  
8212 Graph Generation: Question Answering with Knowledge Base”. In: *Proceedings of the 53rd Annual Meeting  
8213 of the Association for Computational Linguistics and the 7th International Joint Conference on Natural  
8214 Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics,  
8215 pp. 1321–1331. DOI: 10.3115/v1/P15-1128. URL: <http://aclweb.org/anthology/P15-1128>.
- 8216 Yuan, Chenhan and Hoda Eldardiry (Nov. 2021). “Unsupervised Relation Extraction: A Variational Autoen-  
8217 coder Approach”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language  
8218 Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics,  
8219 pp. 1929–1938. DOI: 10.18653/v1/2021.emnlp-main.147. URL: <https://aclanthology.org/2021>  
8220 .emnlp-main.147.
- 8221 Zelenko, Dmitry, Chinatsu Aone, and Anthony Richardella (Mar. 2003). “Kernel Methods for Relation Ex-  
8222 traction”. In: *The Journal of Machine Learning Research* 3, pp. 1083–1106. ISSN: 1532-4435. URL: <https://www.jmlr.org/papers/volume3/zelenko03a/zelenko03a.pdf>.
- 8223
- 8224 Zemlyachenko, Viktor N, Nickolay M Korneenko, and Regina I Tyshkevich (1985). “Graph isomorphism prob-  
8225 lem”. In: *Journal of Soviet Mathematics* 29.4, pp. 1426–1481.
- 8226 Zeng, Daojian, Kang Liu, Yubo Chen, and Jun Zhao (Sept. 2015). “Distant Supervision for Relation Extrac-  
8227 tion via Piecewise Convolutional Neural Networks”. In: *Proceedings of the 2015 Conference on Empirical  
8228 Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics,  
8229 pp. 1753–1762. DOI: 10.18653/v1/D15-1203. URL: <https://aclanthology.org/D15-1203>.
- 8230 Zhao, Yi, Huaiyu Wan, Jianwei Gao, and Youfang Lin (Nov. 2019). “Improving Relation Classification by  
8231 Entity Pair Graph”. In: *Proceedings of The Eleventh Asian Conference on Machine Learning*. Ed. by  
8232 Wee Sun Lee and Taiji Suzuki. Vol. 101. Proceedings of Machine Learning Research, pp. 1156–1171. URL:  
8233 <https://proceedings.mlr.press/v101/zhao19a.html>.
- 8234 Zhou, GuoDong, Jian Su, Jie Zhang, and Min Zhang (June 2005). “Exploring Various Knowledge in Relation  
8235 Extraction”. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.  
8236 Ann Arbor, Michigan: Association for Computational Linguistics, pp. 427–434. DOI: 10.3115/1219840.1  
8237 219893. URL: <https://aclanthology.org/P05-1053>.
- 8238 Zhu, Hao, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun (July 2019). “Graph Neural Net-  
8239 works with Generated Parameters for Relation Extraction”. In: *Proceedings of the 57th Annual Meeting of  
8240 the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics,  
8241 pp. 1331–1339. DOI: 10.18653/v1/P19-1128. URL: <https://aclanthology.org/P19-1128>.
- 8242 Zhu, Xiaojin and Zoubin Ghahramani (2002). “Learning from labeled and unlabeled data with label propa-  
8243 gation”. In: *Technical Report CMU-CALD*. URL: [https://mlg.eng.cam.ac.uk/zoubin/papers/CMU-CALD-](https://mlg.eng.cam.ac.uk/zoubin/papers/CMU-CALD-02-107.pdf)  
8244 02-107.pdf.
- 8245
- 8246
- 8247
- 8248
- 8249
- 8250
- 8251
- 8252
- 8253
- 8254
- 8255
- 8256
- 8257
- 8258
- 8259
- 8260
- 8261
- 8262

8263  
8264  
8265  
8266  
8267  
8268  
8269  
8270  
8271  
8272  
8273  
8274  
8275  
8276  
8277  
8278  
8279  
8280  
8281  
8282  
8283  
8284  
8285  
8286  
8287  
8288  
8289  
8290  
8291  
8292  
8293  
8294  
8295  
8296  
8297  
8298  
8299  
8300  
8301  
8302  
8303  
8304  
8305

#### 8306 COLOPHON

8307 This document is written in Lua<sup>A</sup>T<sub>E</sub>X using PGF/TikZ and PGFPLOTS for figures. Most of the text and math  
8308 are typeset in Latin Modern, while EB Garamond is used for titles. A small amount of characters are from the  
8309 T<sub>E</sub>X Gyre Bonum and XITS fonts. Greek words are typeset in the Greek Font Society's Didot Classic, while  
8310 Chinese excerpts are in the I.Ming font. Finally, the word "THÈSE" on the title page comes from a vectorization  
8311 of Auguste Boulanger's Ph.D. theses (1897).

8312  
8313 The manuscript and sources are freely available at <https://esimon.eu/PhD>.

8314  
8315  This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.  
8316 To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.