Revisiting Unsupervised Relation Extraction

Thy Thy Tran[1], **Phong Le**[1], **Sophia Ananiadou**[1,2]
[1]National Centre for Text Mining, University of Manchester, United Kingdom
[2]The Alan Turing Institute, London, United Kingdom
{thy.tran, phong.le, sophia.ananiadou}@manchester.ac.uk

Abstract

Unsupervised relation extraction (URE) extracts relations between named entities from raw text without manually-labelled data and existing knowledge bases (KBs). URE methods can be categorised into generative and discriminative approaches, which rely either on hand-crafted features or surface form. However, we demonstrate that by using only named entities to induce relation types, we can outperform existing methods on two popular datasets. We conduct a comparison and evaluation of our findings with other URE techniques, to ascertain the important features in URE. We conclude that entity types provide a strong inductive bias for URE.[1]

1 Introduction

Relation extraction (RE) extracts semantic relations between entities from plain text. For instance, "**Jon Robin Baitz**$_{head}$, born in **Los Angeles**$_{tail}$ ..." expresses the relation */people/person/place_of_birth* between the two head-tail entities. Extracted relations are then used for several downstream tasks such as information retrieval (Corcoglioniti et al., 2016) and knowledge base construction (Al-Zaidy and Giles, 2018). RE has been widely studied using fully supervised learning (Nguyen and Grishman, 2015; Miwa and Bansal, 2016; Zhang et al., 2017, 2018) and distantly supervised approaches (Mintz et al., 2009; Riedel et al., 2010; Lin et al., 2016).

Unsupervised relation extraction (URE) methods have not been explored as much as fully or distantly supervised learning techniques. URE is promising, since it does not require manually annotated data nor human curated knowledge bases (KBs), which are expensive to produce. Therefore, it can be applied to domains and languages where annotated data and KBs are not available. Moreover, URE can discover new relation types, since it is not restricted to specific relation types in the same way as fully and distantly supervised methods. One might argue that Open Information Extraction (OpenIE) can also discover new relations. However, OpenIE identifies relations based on textual surface information. Thus, similar relations with different textual forms may not be recognised. Unlike OpenIE, URE groups similar relations into clusters. Despite these advantages, there are only a few attempts tackling URE using machine learning (ML) (Hasegawa et al., 2004; Banko et al., 2007; Yao et al., 2011; Marcheggiani and Titov, 2016; Simon et al., 2019).

Similarly to other unsupervised learning tasks, a challenge in URE is how to evaluate results. Recent approaches (Yao et al., 2011; Marcheggiani and Titov, 2016; Simon et al., 2019) employ a widely used data generation setting in distantly supervised RE, i.e., aligning a large amount of raw text against triplets in a curated KB. A standard metric score is computed by comparing the output relation clusters against the automatically annotated relations. In particular, the NYT-FB dataset (Marcheggiani and Titov, 2016) which is used for evaluation, has been created by mapping relation triplets in Freebase (Bollacker et al., 2008) against plain text articles in the New York Times (NYT) corpus (Sandhaus, 2008). Standard clustering evaluation metrics for URE include B^3 (Bagga and Baldwin, 1998), V-measure (Rosenberg and Hirschberg, 2007), and ARI (Hubert and Arabie, 1985).

Although the above mentioned experimental setting can be created automatically, there are three challenges to overcome. Firstly, the development and test sets are silver, i.e., they include noisy labelled instances, since they are not human-curated. Secondly, the development and test sentences are part of the training set, i.e., a transductive setting.

[1]Source code is available at https://github.com/ttthy/ure

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7498–7505

It is thus unclear how well the existing models perform on unseen sentences. Finally, NYT-FB can be considered highly imbalanced, since only 2.1% of the training sentences can be aligned with Freebase's triplets. Due to the noisy nature of silver data (NYT-FB), evaluation on silver data will not accurately reflect the system performance. We also need unseen data during testing to examine the system generalisation. To overcome these challenges, we will employ the test set of TACRED (Zhang et al., 2017), a widely used manually annotated corpus. Regarding the imbalanced data, we will demonstrate that in fact around 60% (instead of 2.1%) of instances in the training set express relation types defined in Freebase.

In this work, we present a simple URE approach relying only on entity types that can obtain improved performance compared to current methods. Specifically, given a sentence consisting of two entities and their corresponding entity types, e.g., PERSON and LOCATION, we induce relations as the combination of entity types, e.g., PERSON-LOCATION. It should be noted that we employ only entity types because their combinations form reasonably coarse relation types (e.g., PERSON-LOCATION covers */people/person/place_of_birth* defined in Freebase). We further discuss our improved performance in §3.

Our contributions are as follows: (i) We perform experiments on both automatically/manually-labelled datasets, namely NYT-FB and TACRED, respectively. We show that two methods using only entity types can outperform the state-of-the-art models including both feature-engineering and deep learning approaches. The surprising results raise questions about the current state of unsupervised relation extraction. (ii) For model design, we show that link predictor provides a good signal to train a URE model (Fig 1). We also illustrate that entity types are a strong inductive bias for URE (Table 1).

2 Methods for URE

The goal of URE is to predict the relation r between two entities e_{head} and e_{tail} in a sentence s. We will describe three recent ML-based methods tackling URE and our own methods. We divide the ML-based methods into two main approaches: generative and discriminative.

2.1 Generative Approach

Yao et al. (2011) extended topic modelling – Latent Dirichlet Allocation (LDA) (Blei et al., 2003) for RE, developing two models, herewith **RelLDA** and **RelLDA1**. In both models, a sentence and an entity pair perform as a document in topic modelling, while a relation type corresponds to a topic. RelLDA uses three features, i.e., the shortest dependency path between two entities and the two entity mentions. RelLDA1 extends RelLDA with five more features, i.e., the entity types, words and part-of-speech tags between the two entities.

2.2 Discriminative Approaches

Marcheggiani and Titov (2016) proposed a discrete-state variational autoencoder (VAE) to tackle URE (herewith **March**). Their model consists of two components: a *relation classifier* and a *link predictor*. The *relation classifier*, which is discriminative, takes entity types and several linguistic features (e.g., dependencies) as input to predict the relation r. The *link predictor* then uses the (soft) predicted relation r to predict the missing entity e_i in a specific position $\{\text{head}, \text{tail}\}$, given the other entity e_{-i}, where if $i = \text{head}$ then $-i = \text{tail}$ and vice versa. In other words, entity prediction, in a self-supervised manner, provides training signals to learn the relation classifier. However, by using only entity prediction, only a few relation types are chosen. They thus used *entropy* over all relations as a regulariser. The maximisation of the *entropy* regulariser ensures the uniform relation distribution and allows more relations to be predicted.

Another discriminative method is by Simon et al. (2019) (herewith **Simon**) which differs from March in the following ways: a) firstly, its relation classifier employs a piece-wise convolutional network (PCNN) using only surface form without requiring hand-crafted features; b) secondly, they replaced *entropy* with two regularisers: L_s (*skewness*), to encourage the relation classifier to be confident in its prediction, and L_d (*dispersion*), to ensure several relation types are predicted over a minibatch. Note that, L_s is equivalent to the negation of the *entropy* used in March.

2.3 Our Methods

We introduce two entity-based methods, herewith **EType** and **EType+**. Our motivation is that entity types are helpful for RE, as mentioned in Zhang et al. (2017) for supervised learning and Ren et al.

(2017) for distant learning. In URE, Yao et al. (2011); Marcheggiani and Titov (2016) also used entity types. We therefore propose EType that induces coarse relation clusters from the entity types. In particular, given two entity types $t_{e_{head}}$, $t_{e_{tail}}$ as input, EType would output their concatenation $t_{e_{head}}$-$t_{e_{tail}}$ as the relation.

One problem with EType is that the number of relation types is determined by the number of entity types. For instance, 4 entity types lead to $4^2 = 16$ relation types. To extract an arbitrary number of relation types, we build a relation classifier that consists of one-layer feed-forward network taking entity type combinations as input:

$$r = \mathbf{FFN}(t_{e_{head}}\text{-}t_{e_{tail}}),$$

where $t_{e_{head}}$-$t_{e_{tail}}$ is the one hot vector of the entity type pair. We then employ the link predictor used in March and the two regularisers used in Simon, to produce a new method, herewith EType+.

3 Experiments and Results

Evaluation metrics We use the following evaluation metrics for our analysis: a) B^3 (Bagga and Baldwin, 1998) used in previous work, which is the harmonic mean of precision and recall for clustering task; b) V-measure (Rosenberg and Hirschberg, 2007), and c) ARI (Hubert and Arabie, 1985) used in Simon et al. (2019). [2] V-measure is analysed in terms of homogeneity and completeness, while ARI measures the similarity between two clusterings. We note that V-measure is sensitive to the dependency between the number of clusters and instances. A relatively small number of clusters compared to the number of instances should be used to maintain the comparability of using V-measure. More precisely, we evaluated V-measure of the trivial homogeneity, where there are only singular clusters (i.e., each instance is its own cluster). The V-measure of the trivial homogeneity on NYT-FB reached 43.77%, which is higher than all the implemented methods in Table 1. Meanwhile, neither B^3 nor ARI encounters this problem.

Datasets We employed NYT-FB for training and evaluation following previous work (Yao et al., 2011; Marcheggiani and Titov, 2016; Simon et al., 2019). Because only 2.1% of the sentences in NYT-FB were aligned against Freebase's triplets, we were concerned whether this dataset contains enough sentences for a model to learn relation types from Freebase. We thus examined 100 randomly chosen instances from 1.86m non-aligned sentences. We found that 61% of them (or 60% of the whole dataset) express relation types in Freebase. This suggests that the NYT-FB dataset can be employed to train a relation extractor. However, there are two further issues when evaluating URE methods on NYT-FB. Firstly, the development and test sets are all aligned sentences without human curation, which means that they include wrong/noisy labelled instances. In particular, we found that 35 out of 100 randomly chosen sentences were given incorrect relations. Secondly, the two validation sets are part of the training set. This setting is obviously not inductive, as it does not evaluate how a model performs on unseen sentences. Therefore, we *additionally evaluate* all methods (except topic modelling) on the test set of TACRED (Zhang et al., 2017), a widely used manually annotated corpus for supervised RE. The statistics of both NYT-FB and TACRED are provided in Appendix A.

Hyper-parameters We examine three models RelLDA1, March, and Simon using the reported hyper-parameters (Yao et al., 2011; Marcheggiani

Model		B^3	V	ARI
NYT-FB				
RelLDA	$c = 10$	29.1	30.0	13.3
RelLDA1		36.9	34.7	24.2
March (L_s+L_d)		37.5	38.7	27.6
Simon		39.4	38.3	**33.8**
EType+		**41.9**	40.6	30.7
March$^\diamond$ (L_s+L_d)	$c = 16$	36.9	37.4	28.1
EType		41.7	**42.1**	30.7
EType+		41.5	41.3	30.5
RelLDA1	$c = 100$	29.6	-	-
March		35.8	-	-
TACRED				
March$^\diamond$ (L_s+L_d)	$c = 10$	31.0	43.8	22.6
Simon$^\diamond$		15.7	17.1	6.1
EType+		43.3	59.7	25.7
March$^\diamond$ (L_s+L_d)	$c = 16$	34.6	47.6	23.2
EType		**48.3**	**64.4**	**29.1**
EType+		46.1	62.0	27.4
March$^\diamond$	$c = 100$	33.13	43.63	20.21

Table 1: Average results (%) across three runs of different models (except the EType) on NYT-FB and TACRED. c indicates the number of clusters in each method. $^\diamond$ indicates our implementation of the corresponding model. We note that all methods were trained on NYT-FB and evaluated on the test set of both NYT-FB and TACRED.

[2] We used sklearn.metrics package to compute V-measure and ARI.

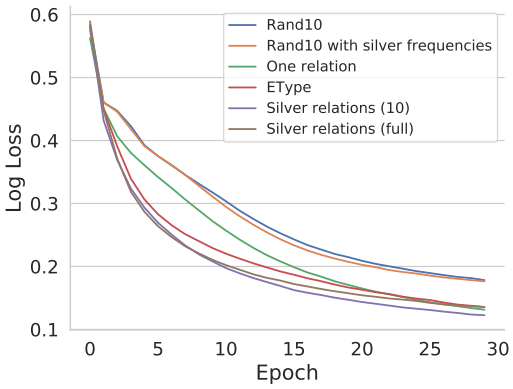


Figure 1: Average negative log likelihood losses across three runs of the link predictor on the training data (not including negative instances). Each line demonstrates a different relation input setting.

and Titov, 2016; Simon et al., 2019). For comparison, we also evaluate March with the two regularisers of Simon, namely **March** ($L_s + L_d$). To evaluate on TACRED, we employed the original March with $n = 100$ using the published repository[3]. Meanwhile, for March (L_s+L_d) and Simon, we reimplemented these models and evaluated them on TACRED. Regarding our methods, EType does not have hyper-parameters, while EType+ uses the same optimiser and entity type dimension as in Simon. All the hyper-parameters used in our experiments are listed in Appendix B.

Results Table 1 demonstrates the average performance of our methods across three runs in comparison with the three ML models on NYT-FB and TACRED. Our models outperform the best performing system of Simon et al. (2019) on both datasets, except ARI on NYT-FB. ARI is shown to be used when there are large equal-sized clusters (Romano et al., 2016) while relation datasets are generally imbalanced (both NYT-FB and TACRED in this study; please refer to Appendix A for the detailed statistics). Due to this reason, ARI might not be appropriate to evaluate URE systems. In addition, the ML methods consistently exhibit lower performance on TACRED than on NYT-FB. The full results are shown in Appendix C.

4 Discussion

The results of our evaluation demonstrate that our models outperform previous methods, despite being simpler than them. These results lead us to the following findings.

Do ML models employ proper inductive biases? In common with other unsupervised learning approaches, there is no guarantee that a URE model would learn the relation types in the used KBs and/or annotated data. A common solution is to employ inductive biases (Wagstaff, 2000) to guide the learning process towards desired relation types. Inductive biases can emanate from pre-processed data. Since our models outperform other methods, we conclude that entity type information alone constitutes a better bias than the biases employed by existing ML models. Indeed, entity types constitute a useful bias for this task. Among the topic modelling based methods, RelLDA1 outperforms RelLDA, which does not employ entity types. In a separate experiment, we found that adding entity types to the Simon model helped to achieve higher performance than the original version, i.e., 42.74% vs. 39.4% F1 B^3 on the NYT-FB test set. However, although both RelLDA1 and March also employ entity types, their performance is still lower than ours. This is because other syntactic and word features used in these two models might cancel out the useful bias of entity types. (More details are in the last paragraph of this section.)

Inductive biases can emanate from training signals. March and Simon are trained from a link predictor, which provides indirect signals to train a relation classifier. Hence, the question here is *"can the link predictor induce good training signals?"* To answer this, we examine the link predictor with alternative settings:

- **Rand10** randomly assigns one among 10 relation types to each entity pair;
- **Rand10 with silver frequencies**, similar to *Rand10*, randomly generates relation types but follows the silver relation distribution;
- **One relation** assumes all entity pairs sharing the same relation type;
- **EType** uses 16 relation types induced from 4 coarse entity types;
- **Silver relations (10)** takes the top 9 most frequent relation types and groups the rest together to form the tenth relation type;
- **Silver relations (full)** considers the full (silver) annotated relations, i.e., 262 types.

Figure 1 illustrates the average loss values of using these settings. If high quality relations were critical for training the link predictor, we would expect lower losses while using annotated relations.

[3] github.com/diegma/relation-autoencoder

Model	B^3	V	ARI
EType+	42.5	40.1	29.2
+Entity	40.5	39.9	28.6
+BOW	37.7	38.0	20.5
+DepPath	41.4	39.4	26.7
+POS	41.6	40.4	27.8
+Trigger	41.7	41.3	29.0
+PCNN	40.8	39.6	27.1

Table 2: Study of EType+ in combination with different features. The results are average across three runs on the development set.

Indeed, the loss curve of using 10 correct relation types is consistently below all the others. This implies that the link predictor is able to provide reasonable signals for training a relation classifier. So why are the Simon and March models outperformed by our models? As pointed out by Simon et al. (2019), the link predictor itself cannot be trained without a good relation classifier. It suggests that the relation classifiers in both methods need to be improved. Empirical evidence shows that both Simon and March models are outperformed (in B^3 and V) by our Etype+, which uses the same link predictor. We also notice that both *One relation* and *EType* at the end sharing similar performances. This might imply that we only need one relation (matrix) to predict head/tail entities, as the link predictor is very expressive. However, the silver relations are clearly helpful as during the first 15 epochs their losses are much lower than others.

Why was the performance on TACRED lower? Despite the fact that TACRED shares similar relation types with Freebase, we observed that both the March and Simon models consistently fare less well in terms of their performance on the TACRED dataset. More precisely, Simon model results in significantly worse performance on TACRED, with 15.7% in terms of B^3, which is twice as low as on NYT-FB (39.4%). This performance drop might be attributed to the distributional shift of the two datasets: variation and semantic shift in vocabulary and language structure over time, since NYT was collected long before TACRED.

How is the performance when combining entity types with other features? Our experiments using only entity types surprisingly perform higher than the previous state-of-the-art methods including feature engineering and deep learning models. However, we know that context information is crucial to distinguish the relation between two entities, as many RE studies have been proposed to integrate the context information to improve the RE performance. We conduct experiments when combining entity types with common features for RE in Table 2. The list of features include: (i) Entity: textual surface form of two entities, (ii) BOW: bag of words between two entities, (iii) DepPath: words on the dependency path between two entities, (iv) POS: part-of-speech tag sequence between two entities, and (v) Trigger: DepPath without stop words. In general, naively combining entity types with other features could not improve the model performance. Additionally, BOW feature had negative effects on the RE performance. This indicates that bag of words between two entities often include uninformative and redundant words, i.e., noises, that are difficult to eliminate using simple neural architectures. While (i)-(v) are widely used handcrafted features for RE, we also incorporated a neural-based context encoder PCNN which is the combination of *Simon*'s PCNN encoder, the entity masking and position-aware attention proposed in (Zhang et al., 2017). However, the performance of combining PCNN is also lower than only entity types.

5 Conclusion

We have shown the importance of entity types in URE. Our methods use only entity types, yet they yield higher performance than previous work on both NYT-FB and TACRED. We have investigated the current experimental setting, concluding that a strong inductive bias is required to train a relation extraction model without labelled data. URE remains challenging, which requires improved methods to deal with silver data. We also plan to use different types of labelled data, e.g., domain specific data sets, to ascertain whether entity type information is more discriminative in sub-languages.

Acknowledgments

We would like to thank the reviewers for their comments, Diego Marcheggiani for sharing his dataset with us, and Étienne Simon for sharing the hyperparameters. The first author thanks the University of Manchester for the Research Impact Scholarship Award. This work is also funded by Lloyds Register Foundation, Discovering Safety Programme, Thomas Ashton Institute.

References

Rabah A Al-Zaidy and C Lee Giles. 2018. Extracting semantic relations for scholarly knowledge base construction. In *2018 IEEE 12th international conference on semantic computing (ICSC)*, pages 56–63. IEEE.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the First Iternational Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, volume 1, pages 563–566.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Francesco Corcoglioniti, Mauro Dragoni, Marco Rospocher, and Alessio Palmero Aprosio. 2016. Knowledge extraction for information retrieval. In *European Semantic Web Conference*, pages 317–333. Springer.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. 2016. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17(1):4635–4666.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387, Florence, Italy. Association for Computational Linguistics.

Kiri Wagstaff. 2000. Refining inductive bias in unsupervised learning via constraints. In *AAAI/IAAI*, page 1112.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh,

Scotland, UK. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A Datasets

Table 3 shows the statistics of the NYT-FB (Marcheggiani and Titov, 2016) and TACRED (Zhang et al., 2017) datasets. We followed the same data split and pre-processing described in Marcheggiani and Titov (2016). For all methods, we trained on NYT-FB and evaluated them on both NYT-FB and TACRED.

Figure 2 illustrates the relation distributions of two datasets: NYT-FB and TACRED. We can see that 15/253 most frequent relations account for 82.97% of the total number of instances in NYT-FB. Meanwhile, 15/41 relations sum upto 74.94% of the total number of instances in TACRED.

B Hyper-parameter Settings

We used the development set to stop the training process. For every model, we conducted three runs with different initialised parameters and computed the average performance. We list the hyper-parameters of different models in Table 4.

C Detailed Results

Table 5 presents the average test scores of three runs on the NYT-FB and TACRED datasets. We note that the two models proposed by Marcheggiani and Titov (2016) and Simon et al. (2019) are sensitive to the hyper-parameters and thus difficult to train. We could not replicate the performance of Simon on the NYT-FB dataset.

	Train	Dev	Test
NYT-FB (#r = 262)			
Raw instances	1,950,557	389,819	1,560,738
Positive	41,685	7,793	33,808
TACRED (#r = 41)			
Raw instances	68,124	22,631	15,509
Positive	13,012	5,436	3,325

Table 3: The statistics of the NYT-FB and the TACRED datasets. #r indicates the number of relation types in each dataset.

Parameter	Ls	$Ls + Ld$
Optimiser	AdaGrad	
Number of epochs	10	
Batch size	100	
L2 regularisation	1e-7	
Feature dimension	10	
Learning rate	0.1	0.005
L_s coefficient	0.1	0.01
L_d coefficient	–	0.02

(a) Marcheggiani and Titov (2016)'s model.

Parameter	**Value**
Optimiser	Adam
Learning rate	0.005
Learning rate annealing	$0.5^{0.25}$
Batch size	100
Early stop patience	10
L2 regularisation	2e-11
Word dimension	50
Entity type dimension	10
L_s coefficient	0.01
L_d coefficient	0.02

(b) Simon et al. (2019)'s model.

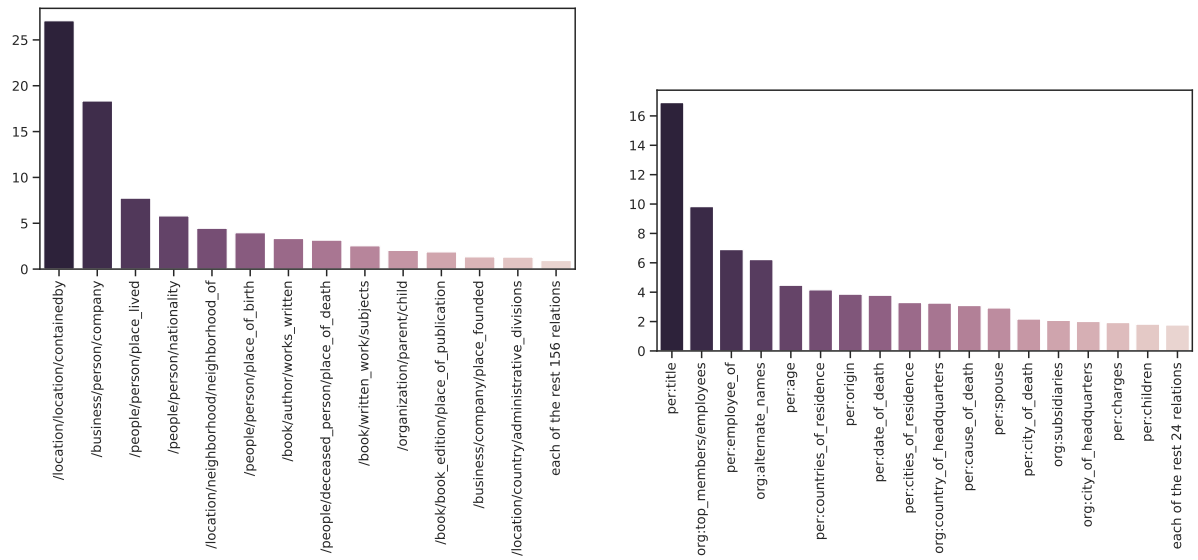
Parameter	**Value**
Optimiser	Adam
Learning rate	0.001
Batch size	100
Early stop patience	10
L2 regularisation	1e-5
Entity type dimension	10
L_s coefficient	0.0001
L_d coefficient	0.02

(c) EType+.

Table 4: Hyper-parameter values used in our experiments.

Model		B^3			V-measure			ARI
		F1	P	R	F1	Hom.	Comp.	
NYT-FB								
RelLDA		29.1	24.8	35.2	30.0	26.1	35.1	13.3
RelLDA1		36.9	30.4	47.0	37.4	31.9	45.1	24.2
March (L_s+L_d)		37.5	31.1	47.4	38.7	32.6	47.8	27.6
March (L_s+L_d)‡	$n = 10$	38.7	30.9	51.7	37.6	31.0	47.7	26.1
Simon		39.4	32.2	50.7	38.3	32.2	47.2	33.8
Simon‡		32.6	28.2	38.9	30.5	26.1	36.8	23.8
EType+		41.9	31.3	63.7	40.6	31.8	56.2	30.7
March (L_s+L_d)‡		36.9	32.0	43.7	37.4	32.6	43.9	28.1
EType	$n = 16$	41.7	32.5	58.0	42.1	34.7	53.6	30.7
EType+		41.5	32.0	59.0	41.3	33.6	53.9	30.5
RelLDA1		29.6	-	-	-	-	-	-
March	$n = 100$	35.8	-	-	-	-	-	-
March‡		34.8	24.4	62.4	25.9	18.7	42.7	13.1
TACRED								
March (L_s+L_d)‡		31.0	21.7	54.9	43.8	35.5	57.2	22.6
Simon‡	$n = 10$	15.7	12.1	22.4	17.1	14.6	20.6	6.1
EType+		43.3	28.0	96.9	59.7	43.4	96.0	25.7
March (L_s+L_d)‡		34.6	24.3	61.3	47.6	38.9	61.4	23.2
EType	$n = 16$	48.3	32.3	96.3	64.4	48.6	95.6	29.1
EType+		46.1	30.3	96.9	62.0	45.8	96.1	27.4
March‡	$n = 100$	33.13	21.83	69.20	43.63	32.96	64.66	20.21

Table 5: Average results (%) across three runs of different models (except the rule-based EType) on two datasets: the distant supervision NYT-FB and the large supervised dataset TACRED. The model of Marcheggiani and Titov (2016) is March and the model of Simon et al. (2019) is Simon. ‡ indicates our implementation of the corresponding model.



(a) NYT-FB has 253 relation types in total

(b) TACRED has 41 relation types in total

Figure 2: Relation distribution of NYT-FB and TACRED (%).