Learning Representations for Relation Classification

Victor Zhong
Department of Computer Science
Stanford University
vzhong@stanford.edu

Abstract

Knowledge bases can be applied to a wide variety of tasks such as search and question answering, however they are plagued by the problem of incompleteness. In this project, we propose two models for automated relation classification using extracted entity pairs and related sentences from natural text. We evaluate both models on a portion of the Stanford KBP dataset across 38 relations, achieving a classification accuracy of 53.65%.

1 Introduction

Knowledge bases such as Freebase [5], NELL [8], and YAGO [18] contain a wealth of information that can be leveraged to improve systems in domains from natural language processing to computer vision. A difficulty in using knowledge bases is that the data contained are incomplete. It may be intractable to construct a knowledge base that is both comprehensive and complete, however it would be helpful if one can leverage existing instances in the knowledge base to make inferences about instances that are not in the knowledge base.

To this end, we propose two approaches to learn distributed representations for knowledge base entries with the goal of enabling such inference. Given a fact in the form of a triplet `subject, relation, object`, we learn representations for each component of the fact to facilitate the prediction of the relation given the entities. Moreover, we leverage the natural language information by jointly learning representations for the sentences in which the two entities are mentioned. In this sense, the model jointly learns the representations for the entities as well as the representation for the context in which they co-occur.

2 Related Work

Embedding based approaches have become a popular method for dealing with tasks such as language modeling [3, 15], statistical machine translation [21], and image captioning [13].

Embedding based approaches have also been applied successfully to knowledge base completion [7, 16, 6, 10]. Another successful technique for knowledge base completion has been proposed by Lao et al [14] using probabilistic random walks. Gardner et al [12] subsequently improved the PRA system by selectively sampling paths according to embedding similarity. In addition, embedding based knowledge base completion techniques have been incorporated into large-scale web-based probabilistic knowledge bases [10, 2].

Embedding based relation classification using natural text was proposed by Socher et Al. [17] using matrix-vector recursive neural networks over parse tree paths between entity pairs. Weston and Bordes proposed a relation extraction model that projects relation mentions in text to the embedding space learned using triplets from the knowledge base [19].

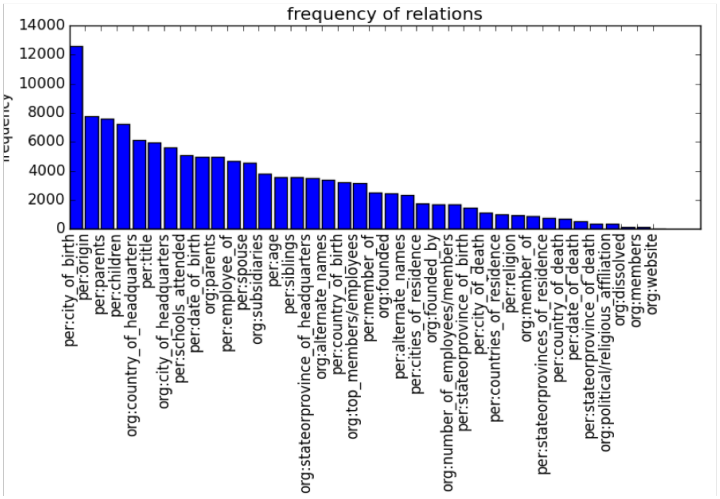

Figure 1: Distribution of training triplets with respect to relation type

The contribution of this work is the incorporation of natural text into the classification task using recurrent models without parsing and without annotated relation mentions. Moreover, we learn this task in a loosely supervised fashion - for each entity pair, we are given a list of known relations between the entity pair as well as a list of sentences in which the entity pair occurs.

3 Approach

3.0.1 Dataset

We use data from Stanford's Knowledge Base Population system as proposed by Angeli et al [1]. This database consists of over 4 million sentences describing 57k unique entities across 38 unique relations concerning people and organizations. There are 130k unique (subject, relation, object) training triplets present in the database. For each entity pair, the database contains a list of known relations between the two entities, as well as a list sentences that mention the two entities.

An example entity pair from the dataset is `Barack_Obama, Michelle_Obama`. A true triplet consisting of this entity pair and known relations is

`Barack_Obama per:spouse Michelle_Obama`

From this dataset, we randomly sample 100k triplets as training data, 20k million triplets as development data, and 10k triplets as test data. The distribution of the triplets with respect to the relation types are shown in figure 1.

In addition to the known relations between entity pairs, we also have a list of sentences in which each entity pair occurs. The relation prediction task is loosely supervised in the sense that a sentence in which the entity pair occurs does not necessarily provide provenance for a valid relation between the two entities.

3.1 Notation

For notational convenience, given a triplet of the form `subject, relation, object`, we denote the subject entity as e_1, the object entity as e_2, and the relation as r. For example, given the triplet

`Barack_Obama per:spouse Michelle_Obama`

e_1 represents the vector representation for `Barack_Obama`, e_2 the vector representation for `Michelle_Obama`, and r the vector representation for `per:spouse`.

3.2 Models

3.2.1 Multilayer Perceptron

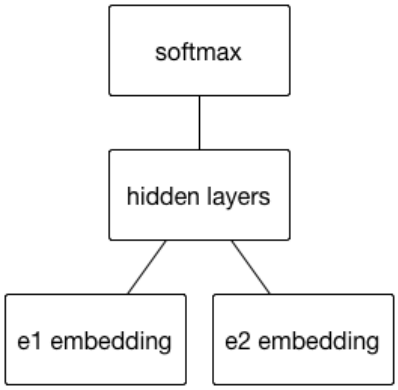

Figure 2: Multilayer perceptron model

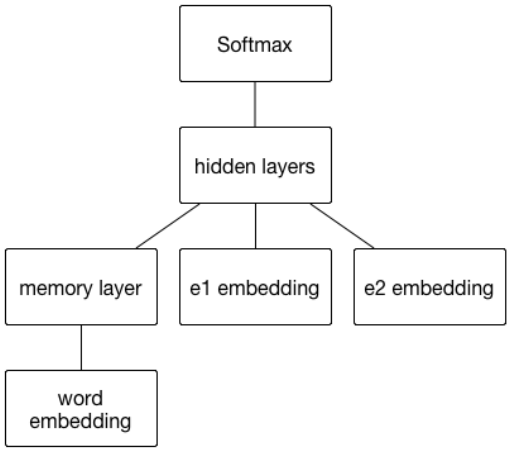

Figure 3: Recurrent model

We begin by introducing a multilayer perceptron model as shown in figure 2.

In this model, the inputs are the distributed representations for the entities involved in the triplet. The concatenated embeddings are projected into a high dimensional space via the hidden layers and condensed into the softmax layer.

During training, we iterate over all unique triplets in the training data, and minimize the categorical cross entropy loss given the entity pairs. This is shown in equation 1.

$$J = -\frac{1}{T}\sum_{i}^{T} \log P(r^{(i)}|e_1^{(i)}, e_2^{(i)}) \tag{1}$$

where T is the number of unique triplets in training, and $e_1^{(i)}$, $e_r^{(i)}$, and $r^{(i)}$ are the respective subject, object, and relation in the ith training triplet.

We train the network via backpropagation, backpropagating the error into the distributed representations of the entities.

3.2.2 Recurrrent Model

The MLP model described in section 3.2.1 does not take into account the context in which the entity pair occurs. In this section, we propose an approach that also models this context by directly utilising the text corpus.

Given an entity pair, we have available the sentences in which the entity pair is contained, as well as the known relations that hold between the entities. An example of this is

```
Rod_Blagojevich graduated from Northwestern University , and
received his law degree from Pepperdine_University , working to
help pay for it .
```

One of the triplets this sentence corresponds to is

```
Rod_Blagojevich per:schools_attended Pepperdine_University
```

Should we know which sentence p provides provenance to which relation, we can construct a model that classifies the relation between the entities given the sentences, as shown in equation 2.

$$J = -\frac{1}{T}\sum_{i}^{T} \log P(r^{(i)}|p^{(i)}) \tag{2}$$

where $p^{(i)}$ is the sentence which provides provenance for the ith triplet.

However, in the database we do not have this correspondence readily available. Instead, for each entity pair, we have available a list of sentences in which the entities occur. With this information available, we attempt to approximate equation 3 as follows:

For each triplet, instead of using the sentence that provides provenance for the triplet, we sample s as a sentence in which the entity pair occurred. This is shown in equation 3. This corresponding model is shown in figure 3. In practice, we found that using only the text snippet between the entity pair yields better performance.

$$J = -\frac{1}{T}\sum_{i}^{T} \log P(r^{(i)}|e_1, e_2, s^{(i)}) \tag{3}$$

where $s^{(i)}$ is a randomly sampled sentence that contains the entity pair present in the ith triplet.

With text information, the recurrent model should be able to distinguish between similar relations such as `per:spouse`, `per:parents`, and `per:children` by learning about the text in which the entity pair occurs. Because the multilayer perceptron model does not take into account this context, we expect it to struggle more with respect to correctly classifying these relations.

4 Experiments

We design and implement our models using Theano [4], a symbolic math library for Python. Unless specified otherwise, we constrain all embedding vectors to have unit norm and use 50 dimensional embedding vectors. We train using Adagrad as described in [11]. For the recurrent model, we use Gated Recurrent Units (GRUs) as described in [9].

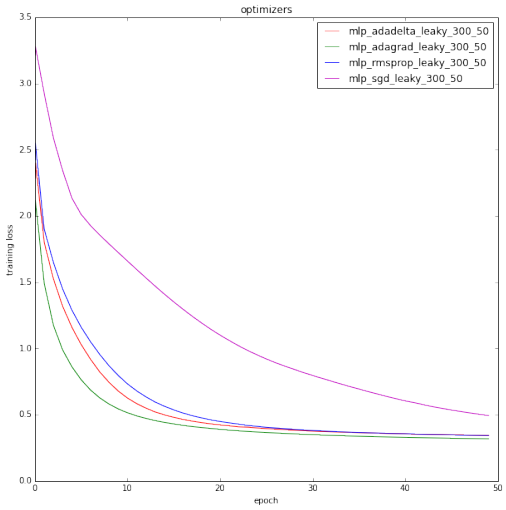

Figure 4: Learning curve across optimization schemes.

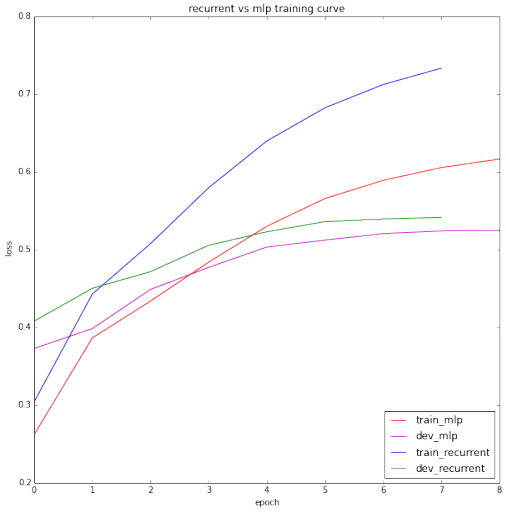

Figure 5: Learning curve for MLP and recurrent model.

We did not find noticeable differences in performance between the tanh, ReLU, and leaky ReLU [20] activations. We were able to achieve faster convergence with Adagrad in comparison to SGD, Adagrad, AdaDelta, and RMSprop optimization as shown in figure 4. The model is a MLP comprised of 100 dimensional embeddings and one 300 dimensional hidden layer.

	MLP	Recurrent
Accuracy	0.5099	0.5365
F1 macro averaged	0.3707	0.3938

Table 1: Test set performance

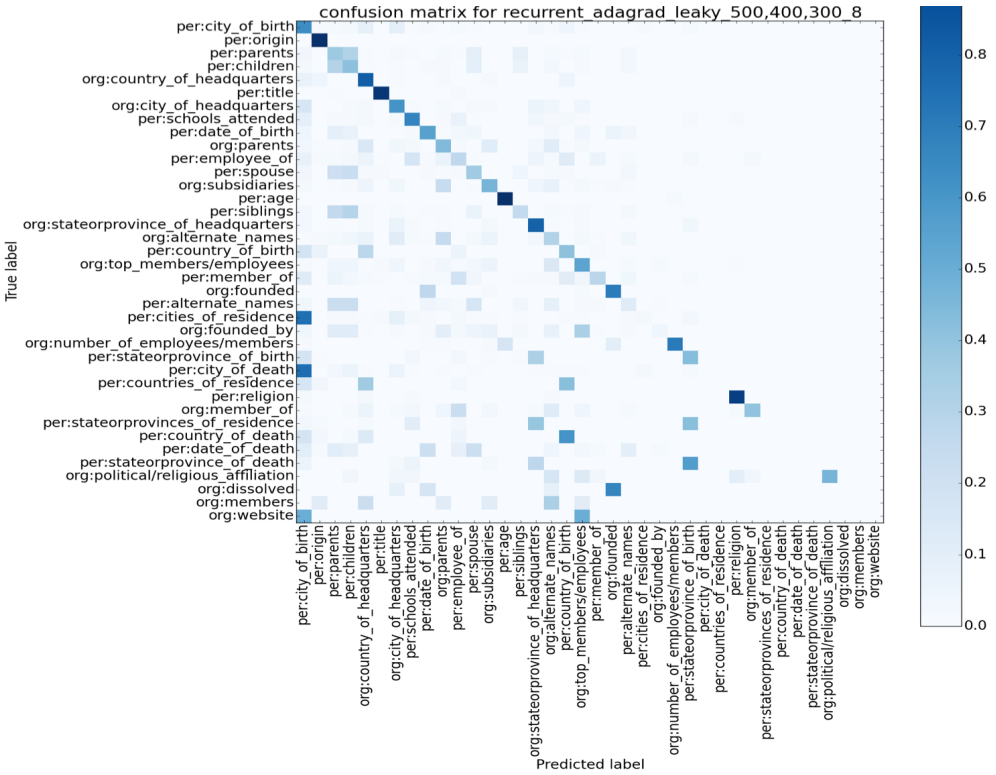

Figure 6: Confusion matrix for recurrent model.

The learning curves for the MLP model and the recurrent model are shown in figure 5. We use 3 hidden layers of dimensions 500, 400, and 300 trained with dropout. We visualize the weights for each relation in appendix A.

We evaluate both models on the test set using multiclass classification accuracy and macro averaged F1. The latter is the calculated using the average of the per-class precisions and the average of the per-class recalls, and hence is noticeably worse due poor performance on the rarer relations (the distribution of the training set across relations is shown in figure 1). The scores are reported in table 1.

The confusion matrix on the test set for the recurrent model is shown in figure 6. We have sorted the columns from the most frequent relations (top-most) to the least frequent relations (bottom-most). We note that the model struggles with rare relations (relations towards the bottom).

Figures 7 and 8 show the confusion matrix for the person-person relations `per:spouse`, `per:siblings`, `per:alternate_names`, `per:parents`, and `per:children` for the MLP model and for the recurrent model. Although both models struggle on these relations, the recurrent model, with the addition of learned representations for the sampled sentence, markedly outperforms the MLP model on these relations. The MLP model frequently confuses person-person relations with `per:parents`, which happens to be the most frequent relation seen during training.

Indeed, these relations are hard to distinguish given that the sampled sentences don't necessarily provide provenance for the corresponding triplet. If the text does not provide enough evidence as to

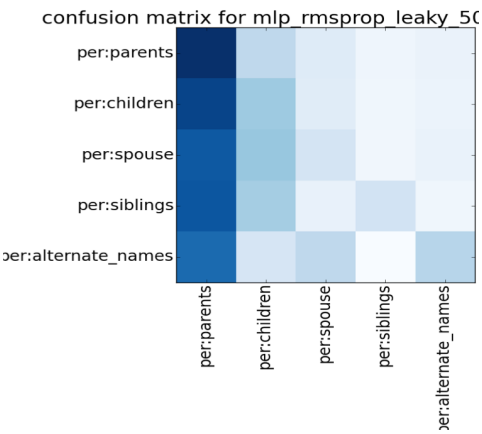

Figure 7: Person-person relation confusion matrix for MLP.

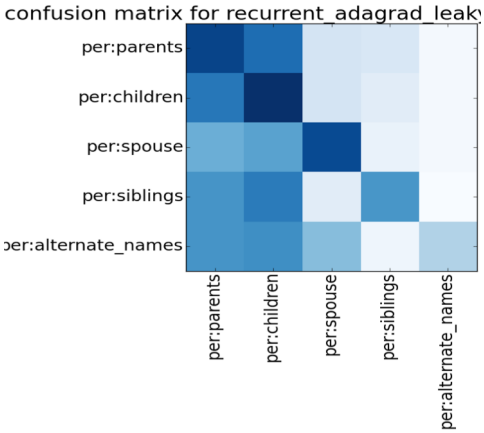

Figure 8: Person-person relation confusion matrix for recurrent model.

which relation is the correct one, the recurrent only has the learned representations of the entity pair to work with. Nevertheless, we note that improved performance on these relations can be achieved even with noisily sampled sentences that may not provide provenance for the triplet.

5 Future Work

One direct extension to this work is to construct a dataset such that each triplet is matched with the corresponding sentence that provide provenance for the triplet. We expect models trained on this dataset to further improve performance on the relation classification task.

One limitation of this work is that the models proposed are not explicitly trained to distinguish mention pairs between which there are valid relations and mention pairs between which there are no valid relations. The ability to do so is crucial for the KBP slotfilling task, whereby the majority of mention pairs extracted from sentences are not related. It is therefore crucial to be able to distinguish between entities that are related and entities that are not related.

We hypothesize that a recurrent model such as the one proposed in this work may be able to distinguish between related entity pairs and unrelated entity pairs given the sentence from which the mention pair is extracted (eg. the provenance for the triplet). While it may be possible to accomplish this task using the proposed model by thresholding the confidence scores, we hypothesize that introducing an explicit objective should be give much better performance. For example, one approach that may prove fruitful is to add to the objective of equation 3 another sigmoid unit to predict whether a relation exists between the two entities.

6 Conclusion

In this project, we proposed two models for automated relation classification: a multilayer perceptron that learns representations for the entity pair, and a recurrent model that additionally learns representations for context in which the entity pair occurs. Despite that the sampled sentences are noisy in that they do not necessarily provide provenance for the given triplet, we show that by leveraging learned representations for this context, the recurrent model is able to distinguish between similar relations much better than the multilayer perceptron model. Finally, we evaluate both models on a portion of the KBP dataset across 38 relations, achieving a classfication accuracy of 53.65%.

References

[1] G. Angeli, A. Chaganty, A. Chang, K. Reschke, J. Tibshirani, J. Y. Wu, O. Bastani, K. Siilats, and C. D. Manning. Stanfords 2013 KBP system. 2014.

[2] G. Angeli and C. D. Manning. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, 2014.

[3] Y. Bengio, H. Schwenk, J.-S. Sencal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. *Innovations in Machine Learning*, pages 137–186, 2006.

[4] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3, 2010.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[7] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning Structured Embeddings of Knowledge Bases. In *AAAI'11*, 2011.

[8] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, volume 5, page 3, 2010.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated Feedback Recurrent Neural Networks. *arXiv:1502.02367 [cs, stat]*, Feb. 2015. arXiv: 1502.02367.

[10] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[12] M. Gardner, P. Talukdar, J. Krishnamurthy, and T. Mitchell. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of EMNLP*, 2014.

[13] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603, 2014.

[14] N. Lao, T. Mitchell, and W. W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[16] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[17] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[18] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[19] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.

[20] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[21] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *EMNLP*, pages 1393–1398, 2013.

A
Relation embeddings

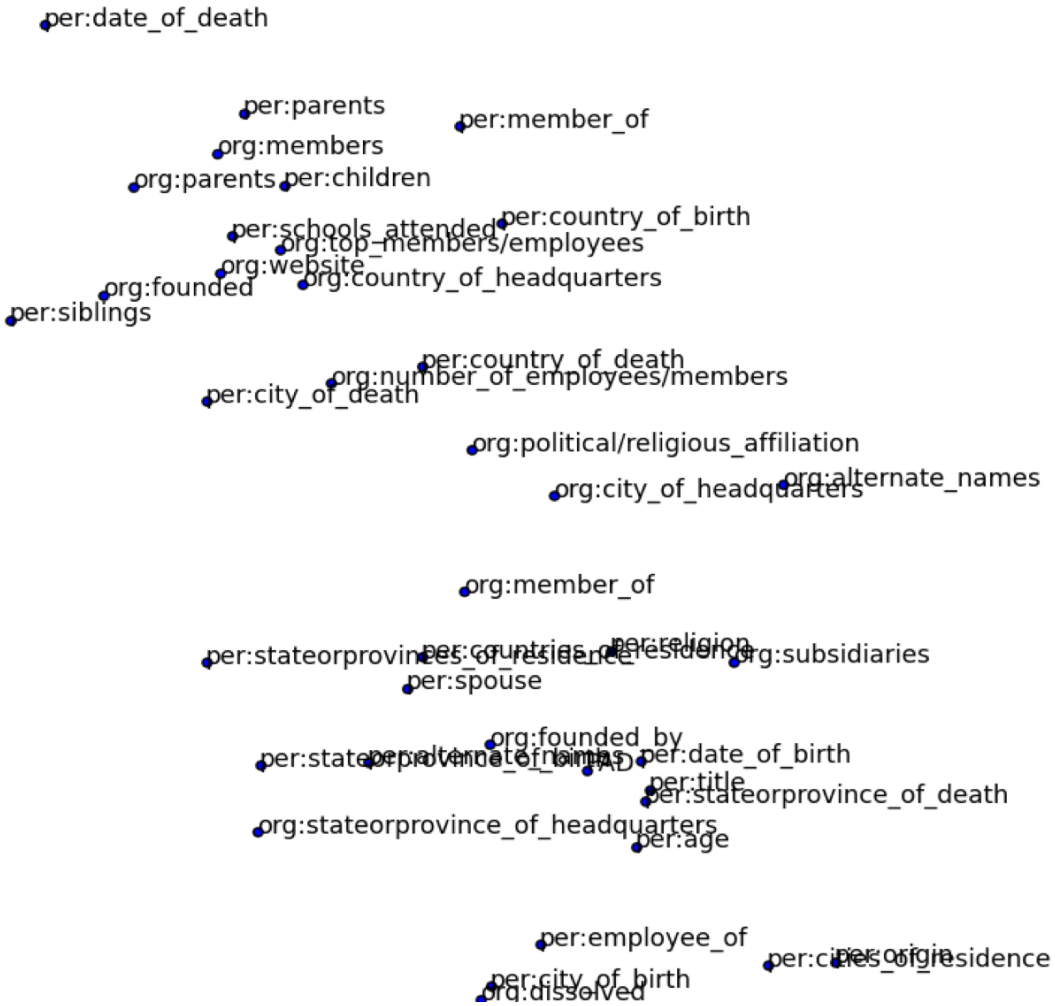

Figure 9: Relation weights reduced via SVD