EMBEDDING ENTITY PAIRS THROUGH OBSERVED RELATIONS FOR KNOWLEDGE BASE COMPLETION

Dirk Weissenborn
Language Technology Lab, DFKI
Alt-Moabit 91c
Berlin, Germany
dirk.weissenborn@dfki.de

ABSTRACT

In this work we present a novel approach for the utilization of observed relations between entity pairs in the task of triple argument prediction. The approach is based on representing observations in a shared, continuous vector space of structured relations and text. Results on a recent benchmark dataset demonstrate that the new model is superior to existing sparse feature models. In combination with state-of-the-art models, we achieve substantial improvements when observed relations are available.

1 INTRODUCTION

The task of automatic knowledge base completion (AKBC) has gained a lot of attention in the past years, due to the creation of large knowledge bases (KBs), such as DBpedia (Auer et al. (2007)) or Freebase (FB, Bollacker et al. (2008)). Typically, they store structured information in form of triples that consist of a relation and its arguments, i.e., subject- and object-entity (entity pair). AKBC aims at adding missing relations between entity pairs using existing information. Such information can be extracted from various sources, like structured knowledge bases and unstructured text, usually in form of direct connections between the entity pair. We call these connections observed relations.

There have been several approaches on solving AKBC, including algorithms exploiting the graph structure of KBs (e.g., path ranking) (Lao et al. (2011)) or latent feature models (Riedel et al. (2013); Yang et al. (2014); Toutanova et al. (2015)). While there has been a lot of work on scoring triples with latent feature models directly, to our knowledge, there has been no attempt on modeling entity-pairs through observed relations using continuous vector representations.

To this end, we developed a new model that represents observed relations between entity pairs in a continuous vector space, which allows sharing of latent features. This leads to improved generalization over the use of sparse feature representations. Results of our experiments demonstrate that this approach is superior to prior work, and contributes a significant improvement on triple argument prediction in combination with existing models, especially when observed relations are present.[1].

2 MODELS

For the descriptions in this section we use the following notation. We consider a KB to be a triple (E, R, T). R refers to the set of all relation types and E to the set of all KB entities. $R = R_{kb} \cup R_{text}$ is the union of all structured knowledge base- and textual relation types (unique dependency paths). T denotes the set of all facts containing triples $(r, s, o) \in T$, $r \in R$, $s, o \in E$. $R^{-1} = \{r^{-1} | r \in R\}$ is the set of inverse relations and analogous $T^{-1} = \{(r^{-1}, o, s) | (r, s, o) \in T\}$ the set of inverse triples. $R_{s,o} = \{r | (r, s, o) \in T \cup T^{-1}\} \cup \{r_{def}\}$ is the set of relations that occur between entities s and o, including a default relation r_{def}. Finally, we use $\mathbf{v}$ to refer to vectors.

[1] Implementation at https://github.com/dirkweissenborn/genie-kb, based on TensorFlow (Abadi et al. (2015)).

Model	Reference	Scoring Function
DistMult (D)	Yang et al. (2014)	$f_D(r, s, o) = \mathbf{v}_r \odot (\mathbf{v}_s \circ \mathbf{v}_o)$
Model E	Riedel et al. (2013)	$f_E(r, s, o) = \mathbf{v}_r^{subj} \odot \mathbf{v}_s + \mathbf{v}_r^{obj} \odot \mathbf{v}_o$
Model F	Riedel et al. (2013)	$f_F(r, s, o) = \mathbf{v}_{(s,o)} \odot \mathbf{v}_r$
Model N	Riedel et al. (2013)	$f_N(r, s, o) = \sum_{r' \in R_{s,o} \setminus \{r\}} w_{r,r'}$
Model O	this work	$f_O(r, s, o) = \sum_{r' \in R_{s,o} \setminus \{r\}} w(r', r, s, o) \cdot \boldsymbol{v'}_{r'} \odot \boldsymbol{v}_r$

Table 1: Existing scoring models, their reference and respective scoring functions. For brevity, we omit the application of tanh to the entity and relation representations before scoring, that we found useful because of regularization effects (see also Yang et al. (2014)).

The presented models are all scoring models. I.e., they define a scoring function $f : R \times E \times E \to \mathbb{R}$, which calculates a score for a given triple. The higher the score, the more likely the triple is considered to be true by the model. See Table 1 for existing models.

Latent Feature Models DistMult, Model E and Model F are three previously introduced models, that try to capture all knowledge within latent entity(-pair) and/or relation type representations. To this end, Model F scores a triple by the compatibility between the triple's entity pair and relation. DistMult is slightly different in that it represents an entity pair as the element-wise multiplication of their individual representations. Finally, Model E calculates the compatibility score between the arguments of a triple with their corresponding argument-position wrt. the relation.

Observed Feature Models Model N is based on the association strength between the triple relation and all other observed relations between the triple entities. Toutanova & Chen (2015) introduced other observed feature models, namely NodeFeat & LinkFeat. However, LinkFeat is the same model as Model N, and NodeFeat can be considered the non-latent varient of Model E, which also performs similarly to Model E (Toutanova & Chen (2015)). Therefore, we do not consider those here.

Model O In contrast to Model N, we model the association strength of a triple relation and observed relations as their pairwise weighted similarity (dot-product) in a shared vector space.

The simplest weighting for Model O is the uniform weighting ($\mathrm{O_u}$):

$$w_u(r', r, s, o) = \frac{1}{|R_{s,o} \backslash \{r\}|} \quad .$$

In reality, however, many observed textual relations between entities are either noise or do not indicate a relation. Since one observation can suffice to indicate a relation, an approach that weighs relation indicating observations stronger might be preferable. This can be achieved with a selective weighting approach ($\mathrm{O_s}$):

$$w_m(r', r, s, o) = \frac{e^{s(r',r)}}{\sum_{r'' \in R_{s,o} \setminus \{r\}} e^{s(r'',r)}} \text{ , where } \; s(r', r) = \boldsymbol{v'}_{r'} \odot \boldsymbol{v}_r \; .$$

The advantages of Model O compared to Model N are: 1) use of shared latent features among observed relations, and 2) vector representations for observed relations can be computed by a composition function, e.g., ConvNN (Toutanova et al. (2015)) or RNN. Both advantages should lead to an improved generalization.

Combined Models The idea of combining different scoring models has been exploited in other works as well Riedel et al. (2013); Toutanova & Chen (2015), however, they were combined using a weighted sum with fixed weights, either uniform or optimized on a validation set via grid search. In this work, we trained those combination weights jointly with the combined models. E.g., $f_{D+E+O}(r, s, o) = f_D(r, s, o) + \alpha_E \cdot f_E(r, s, o) + \alpha_O \cdot f_O(r, s, o)$. We initialize all α with 1.

Training-Loss We optimize the per triple cross-entropy loss on the softmax of the positive and 200 sampled, corrupted (negative) triple scores. The corruption process closely follows the methodology of Toutanova & Chen (2015) that employs type constraints during negative sampling. Corrupted triples are sampled for each given positive triple and argument position (s and o).

	MRR			HITS@10		
Model	`a`	`t`	`nt`	`a`	`t`	`nt`
N	-	22.1	-	-	32.7	-
O$_u$	-	26.2	-	-	**51.4**	-
O$_s$	-	**29.5**	-	-	47.2	-
D+E	28.2	28.7	**28.0**	45.4	45.7	**45.3**
D+E+N	27.4	26.1	27.9	43.5	43.1	43.6
D+E+O$_u$	30.0	**40.0**	26.1	46.6	**63.2**	40.1
D+E+O$_s$	**30.9**	38.8	27.8	**48.7**	57.7	45.1

Table 2: MRR (scaled by 100) and HITS@10 (in %) performance of triple argument prediction on FB15k-237. `a` - results for all test triples, `t` - triples with entity pairs having textual mentions, `nt` - triples with entity pairs having no mentions.

3 Experiments

We experimented with the FB15k-237 dataset (Toutanova et al. (2015)), which consists of 272,115/17,535/20,466 train/validation/test triples derived from FB and about 4M textual triples from the annotated ClueWeb12 corpus Gabrilovich et al. (2013).

Mini-batch ($B = 1,000$ positive examples) training with ADAM ($\beta_1 = 0.0, \beta_2 = 0.999$, learning rate 0.01)(Kingma & Ba (2014)) is applied for all models. We randomly sample with probability τ a batch of textual triples or else FB triples. We perform a grid search on various hyper-parameter settings.[2]. Early stopping is done base on the MRR (Mean reciprocal rank) on the validation set.

For validation and evaluation, we follow the setup of Toutanova & Chen (2015). We report the mean reciprocal rank (MRR) and HITS@10[3] of scored test triples that are ranked together with their respective negative triples.

4 Results

We compared the performance of models that exploit observations for triple scoring, namely Model O and N, and their combinations with model DistMult (D) and E, namely D+E, D+E+N and D+E+O.[4] The focus lies on column `t` (28.1% of `a`), because only triples of `t` are directly affected by Model O and N. Results are presented in Table 2.

The results clearly show the superiority of Model O compared to N, with large improvements on `t`. Combining Model N with D and E even hurts performance slightly in our setup, which we believe stems from over-fitting on sparse observations. In contrast, Model O in combination with D and E improves performance especially for `t`, where improvements are substantial. However, there is a slight decrease in performance on `nt`. We believe this is due to the strong performance of Model O on training triples with textual mentions, for which the other models do not need to learn additional entity related information. Nevertheless, model D+E+O$_s$ shows the best overall performance.

Another interesting finding is that the two variations of Model O show mixed results. O$_s$ alone performs better in terms of MRR, but O$_u$ shows better results on HITS@10. We believe that O$_s$ captures the connection between very indicative observations and their corresponding relations better than O$_u$. On the other hand, O$_u$ seems to provide better generalization, since all observations (in contrast to only the most likely as in O$_s$) are trained to have representations similar to their corresponding triple relation.

5 Conclusion

In this paper we presented a novel approach for scoring knowledge base triples by exploiting observed relations between entity pairs. It is based on representing observed relations in a shared vector space. Results demonstrate that it is superior to sparse feature models with substantial improvements when textual mentions are observed. Future directions involve modeling representations of observed relations through composition functions to further improve generalization.

[2] $\tau \in \{0.4, 0.6, 0.8\}$ ($\tau = 0.935$ corresponds to Text/FB dataset ratio); representation dim. $\in \{10, 20, 40\}$

[3] fraction of test triples ranked within top 10

[4] Only textual triples are used as observed features.

ACKNOWLEDGMENTS

This research was partially supported by the German Federal Ministry of Education and Research (BMBF) through the projects ALL SIDES (01IW14002), BBDC (01IS14013E), and by the German Federal Ministry of Economics and Energy (BMWi) through the project SD4M (01MD15007B).

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

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

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. Facc1: Freebase annotation of clueweb corpora, 2013.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ni Lao, Tom Mitchell, and William 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*, pp. 529–539. Association for Computational Linguistics, 2011.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. 2013.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. *ACL Association for Computational Linguistics*, 2015.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.