Orthogonality regularizer for question answering

Chunyang Xiao, Guillaume Bouchard, Marc Dymetman, Claire Gardent

Orthogonality regularizer for question answering

Chunyang Xiao[1], Guillaume Bouchard[2], Marc Dymetman[1],Claire Gardent[3]

[1]Xerox Research Centre Europe, Grenoble, France
[2]University College London, United Kingdom
[3]CNRS, LORIA, Nancy, France
[1]chunyang.xiao, marc.dymetman@xerox.com
[2]g.bouchard@cs.ucl.ac.uk
[3]claire.gardent@loria.fr

Abstract

Learning embeddings of words and knowledge base elements is a promising approach for open domain question answering. Based on the remark that relations and entities are distinct object types lying in the same embedding space, we analyze the benefit of adding a regularizer favoring the embeddings of entities to be orthogonal to those of relations. The main motivation comes from the observation that modifying the embeddings using prior knowledge often helps performance. The experiments show that incorporating the regularizer yields better results on a challenging question answering benchmark.

1 Introduction

Having a system which is able to answer questions based on a structured knowledge base is a challenging problem. The problem has been addressed recently by researchers working on large knowledge bases such as Reverb (Fader et al., 2011) and Freebase (Bollacker et al., 2008). The creation of question answering (QA) benchmarks for these knowledge bases (KB) has a significant impact on the domain, as shown by the number of QA systems recently proposed in the literature (Berant and Liang, 2014; Berant et al., 2013; Bordes et al., 2014a; Bordes et al., 2014b; Fader et al., 2013; Fader et al., 2014; Yao and Van Durme, 2014; Yih et al., 2014; Dong et al., 2015).

We identify two types of approaches for KB-centric QA systems: parsing-based approaches and information retrieval (IR) based approaches. Parsing-based approaches (Yih et al., 2014; Berant et al., 2013; Berant and Liang, 2014; Reddy et al., 2014) answer factoid questions by learning a structured representation for the sentences, called logical form. This logical form is then used to query the knowledge base and retrieve the answer. IR-based approaches try to identify the best possible match between the knowledge base and the question (Bordes et al., 2014a; Bordes et al., 2014b; Yao and Van Durme, 2014; Dong et al., 2015). In this work, we focus on the second approach, using embedding models, mainly because it is robust to invalid syntax and can exploit information of the answer.

We focus on the Wikianswers (Fader et al., 2013) dataset constructed for Reverb. On Wikianswers, the underlying semantics is very simple (just one single triple). However, the task remains challenging due to the large variety of lexicalizations for the same semantics. We follow the approach of Bordes et .al (2014b) which learns the embeddings of words and KB elements. They model the semantics of natural language sentences and KB triples as the sum of the embeddings of the associated words and KB elements respectively. Despite its simplicity, this model performs surprisingly well in practice. Something even more interesting (Bordes et al., 2014b) is that the system can have a good performance even without using a paraphrase corpus. This makes the system very attractive in practice because in many specific domains, we might have a KB but there may be no paraphrase corpus as in Wikianswers.

In our work, we push the results further when learning a QA system based only on the KB. Our contribution is to introduce a new orthogonality regularizer which distinguishes entities and relations. We also investigate the tradeoff captured by the orthogonality constraints. With a synthetic example, we show that if entities and relations are independent, orthogonal embeddings generate better results. The orthogonality constraint in the context of question answering is new, although it has been successfully used in other contexts (Yao et al., 2014). Like (Bordes et al., 2014b), we use al-

most no linguistic features such as POS tagging, parsing, etc.

2 The ReVerb Question Answering Task

The ReVerb question answering task was first introduced in (Fader et al., 2013) as follows. Given a large RDF KB and a natural language (NL) question whose answer is given by a triple contained in that KB, the task is to find a correct triple. For example, a correct answer to the NL question "What is the main language in Hong Kong ?" would be the KB triple *(cantonese.e, be-major-language-in.r, hong-kong.e)*. RDF triples are assertions of the form (e_1, r, e_2) where r is a binary relation from some vocabulary R and e_1, e_2 are entities from a vocabulary E.

The KB used is ReVerb[1], a publicly available set of 15 million extractions (Fader et al., 2011) defined over a vocabulary of approximately 600K relations and 3M entities. The test set used for evaluation includes 698 questions extracted from the website Wikianswers, many of which involve paraphrases.

3 Related Work

Fader et al. (2013) present one of the first approaches for dealing with open domain question answering. To map NL questions to KB queries, they first induce a lexicon mapping NL expressions to KB elements using manually defined patterns, alignments and a paraphrase corpus. Using this lexicon, multiple KB queries can be derived from a NL question. These queries are then ranked using a scoring function.

Bordes et al. (2014b) introduce a linguistically leaner IR-based approach which identifies the KB triple most similar to the input NL question. In their approach, KB triples and NL questions are represented as sums of embeddings of KB symbols and words respectively. The similarity between a triple and a question is then simply the dot product of their embeddings. Interestingly, Bordes' (2014b) system performs relatively well (MAP score 0.34) on the Wikianswers dataset even without using the paraphrase corpus. This suggests that the embedding method successfully captures the similarity between NL questions and KB queries. Our work continues this direction by further separating relations with entities.

[1] `http://reverb.cs.washington.edu`

The idea of distinguishing entities and relations in question answering can also be found in (Yih et al., 2014). However, they base their work by supposing that we can cut the sentence into "entity part" and "relation part" and then calculate the matching score. Our model does not need this cut and simply enforces the entity embeddings and relation embeddings (on the KB side) to be different.

Orthogonality or near orthogonality is a property which is desired in many embedding techniques. In random indexing (Sahlgren, 2005), a near orthogonality is ensured amongst the embeddings of different contexts. In (Zanzotto and Dell'Arciprete, 2012), to approximate tree kernels in a distributed way, different subtree feature embeddings are also constructed to be near orthogonal.

Our work gives yet another motivation for orthogonal embeddings for the special case where the semantics of a sentence is modeled as the sum of its associated word embeddings. In this case, orthogonal word embeddings help to model their independence.

4 Embedding model

Word embeddings are generally learned (Deerwester et al., 1990; Mikolov et al., 2013; Lebret and Collobert, 2015; Faruqui et al., 2014) such that words with similar context will naturally share similar embeddings as measured for instance by cosine similarity. The embeddings learned in (Bordes et al., 2014b) also encode context information. They link the embedding of words with the whole triple-answer in their scoring function. By this means, the word embedding carries the information of the whole triple.

Our model further distinguishes entities and relations. Noting that entities and relations may have some independence (knowing that 'a man eats' doesn't help to tell 'which man'), the distinction is done via orthogonality. We show in the toy example that orthogonality helps to capture this independent structure of the data.

4.1 Scoring function

The model learns the embedding of each word and KB element by trying to score the correct answers highest. Mathematically, let q be the query, and a be the answer-triple to align. Denote the total number of words as N_w and the number of KB elements as N_{kb}. Then denote by $\phi(q) \in \{0,1\}^{N_w}$

Algorithm 1 Training with orthogonality regularizer

1. Sample a positive training pair (q_i, a_i) from D.
2. Create a corrupted triple a_i'
3. If $S(q_i, a_i) - S(q_i, a_i') < 0.1$:
   make a stochastic gradient ascent on $S(q_i, a_i) - S(q_i, a_i') - \lambda|E.R|$
4. Normalize the embedding vector

the 1-hot representation indicating the presence or absence of words in the query. Similarly we denote the sparse representation on the KB side as $\psi(a)$. Let $M \in R^{d \times N_w}$ be the embedding matrix for words and $K \in R^{d \times N_{kb}}$ be the embedding matrix for the elements in the KB. d is the low dimension chosen by the user.

The embedding of the sentence is then calculated as $M\,\phi(q)$ and similarly the embedding of the answer-triple as $K\,\psi(a)$. We can score the matching of these embeddings:

$$S(q, a) = (M\,\phi(q))^\top (K\,\psi(a))$$

which is the dot product between the embedding of the sentence and the embedding of the triple. The model is introduced in (Bordes et al., 2014b) and we use the same scoring function. Note that the model actually sums up each word embedding to form the embedding of the sentence.

4.2 Inference

The inference procedure is straightforward. Given a question q and a set of possible answer triples noted $A(q)$, the model predicts the answer by returning the triple with the highest score:

$$a' = argmax_{a \in A(q)} S(q, a)$$

4.3 Training

Originally in (Bordes et al., 2014b), given a question to be answered, training is performed by imposing a margin-constraint between the correct answer and negative ones. More precisely, note a' a negative answer to the question q (the correct answer to q being a). Then for each question answer pair, the system tries to maximize the following function by performing a gradient ascent step:

$$min(\epsilon,\ S(q, a) - S(q, a'))$$

with ϵ the margin set to 0.1. In addition, the norms of columns in M and K are constrained to be inferior to 1. The training is done in a stochastic way by randomly selecting a question answer pair at each step. For each gradient step, the step size is calculated using Adagrad (Duchi et al., 2011). The negative example is created by randomly replacing each element of (e_1, r, e_2) by another one with probability 2/3.

4.4 Enforcing Orthogonal Embeddings

In this work, we are especially interested in the additional assumptions we can make on the model in order to cope with data sparsity. Indeed, when the number of training data supporting the computation of embeddings is small, embedding models are brittle and can lead to disappointing results. We noticed that one important assumption that is not discussed in the basic approach is that the embedding space is the same for relations and entities. That approach has a tendency to learn similar embeddings for entities and relations, even if they have different meanings. Intuitively, we would like to balance that tendency by a "prior knowledge" preference towards choosing embeddings of entities and relations which are orthogonal to each other.

To justify this assumption, consider a simple case where the underlying semantics is (e, r) as in the sentence "John eats". We will use the same letter to indicate an entity or relation and their corresponding embeddings. In (Bordes et al., 2014b), the embedding of the semantics is then calculated as $e + r$ for this very simple case. Now suppose that $\forall e' \neq e,\ ||e - e'||_2 \geq \epsilon$ (i.e John is different from Mary with margin ϵ) and that the same kind of constraints also holds for relations. However, even when these constraints are satisfied, it is not guaranteed that $||e + r - e' - r'||_2 \geq \epsilon$, which means that the model may still get confused on the whole semantics even if each part is clear.

One obvious and linguistically plausible solution is to say that the entities and relations lie in orthogonal spaces. Indeed, if relations and entities are orthogonal ($\forall r, e\ (r\ \perp\ e)$), then if two entities e, e' and two relations r, r' are distinct (i.e., $||e - e'||_2 \geq \epsilon$ and $||r - r'||_2 \geq \epsilon$), it follows that $||e + r - e' - r'||_2 = ||e - e'||_2 + |||r - r'||_2 \geq 2\epsilon$ by Pythagorean theorem. That is, two sentences whose semantic representations involve two distinct entities and/or relations will have different values.

In real problems, however, posing a hard orthogonality constraint largely reduces the model's

sentence	*Embedding*	*This work*
What is the argument on gun control ?	*(short-gun.e be-type-of.r gun.e)*	***(giuliani.e support.r gun-control.e)***
What year did minnesota become part of US ?	***(minnesota.e become-state-on.r may-11-1858.e)***	*(minnesota.e be-part-of.r united-states.e)*
What is the religious celebration of christians ?	*(christian.e be-all-about.r original-sin.e)*	***(easter.e be-most-important-holiday.r christian.e)***
What do cassava come from ?	*(cassava.e be-source-of.r security.e)*	***(cassava.e be-grow-in.r africa.e)***

Table 1: Some examples for which our system differs from ((Bordes et al., 2014b)). Gold standard answer triples are marked in bold.

expressive power[2], so we decide to add it as a regularizer. More concretely, let the correct triple be (e_1, r, e_2) and the negative one be (e'_1, r', e'_2). Consider that we are in a case not satisfying the margin constraint, then we will try to maximize the following regularized function $S(q, a) - S(q, a') - \lambda|E.R|$ with a gradient step. The regularizer $|E.R| = |e_1.r| + |e_2.r| + |e'_1.r'| + |e'_2.r'|$ is minimized when all the entities and relations live in orthogonal space. The regularization parameter λ is chosen via an automatically constructed development set for which we randomly selected 1/2000 of all the triples in the KB and generate associated questions. We discard these triples from training and choose the λ value based on the score on the development set. The λ value is by this means set to 0.01 with λ in $\{0.5,0.1,0.05,0.01,0.005,0.001\}$. Once the λ value is chosen, we retrain the whole system.

5 Experimental results

5.1 Toy example

In this section, we illustrate the benefits of orthogonality via a toy example. We construct a KB containing 50 entities (E) and 50 relations (R) then generate all their cross products obtaining 2500 fact pairs. In consequence the entities and relations are independent.

For every $e_i \in E$, we suppose that there is a single word lexicalizing the entity noted "e_i" . Similarly, we note the lexicalization of r_j "r_j". We separate these 2500 pairs into training (2450) and test (50). Notice that similarly to Wikianswers, this toy dataset involves KB entities and relations whose type is known *a priori*.

The training corpus is built using one simple generation rule : $(e_i, r_j) \rightarrow$ "$e_i\ r_j$" . Negative examples are created by replacing with probability 1/2 both entity and relation with another one. We embed all the words and KB symbols in a space of 20 dimensions. We compare the model (Bordes et al., 2014b) with the model where we enforce E and R (and also "E" and "R") to be orthogonal. This means that words or KB symbols in fact live in an embedding space of dimension 10.

Model	Accuracy (1)	Accuracy (2)
Embedding	76%	54%
Ortho_Embedding	90%	68%

Table 2: Results on toy example.

At test time, for a given sentence "$e_i\ r_j$", a set of (e, r) pairs is ranked and we compute the proportion of cases where the first ranked pair is correct. Table 2 shows the results for both systems on two configurations: a configuration (Accuracy(1)) where the number of pairs to be ranked is 1250 and another (Accuracy(2)) with 2500 pairs.[3] In both cases, imposing the orthogonality constraint improves performance by a large margin.

5.2 Wikianswers

Wikianswers contains a set of possible triples for each question and we re-rank these triples to report our system's performance. This is the "re-ranking" setting used in (Bordes et al., 2014b). Table 3 compares different systems in this setting. The Embedding scores are taken from (Bordes et al., 2014b) for which we have reimplemented and confirmed the results.

Method	Prec	Recall	F1	MAP
Embedding	0.60	0.60	0.60	0.34
This work	0.63	0.63	0.63	0.36

Table 3: Performance for re-ranking question answer pairs of test set for different systems on Wikianswers

Table 3 shows that our technique improves the performance also on the larger, non-synthetic,

[2]Especially, if the embeddings are orthogonal between entities and relations, the knowledge of a given entity can not help to infer the relation and vice versa.

[3]We make sure that the correct answer is included .

dataset provided by Fader (2013) over the Bordes (2014b)'s method. In addition, Table 1 shows some examples where the two systems differ and where the orthogonality regularized embeddings seem to better support the identification of similar relations. For instance, "is the argument on" is mapped to *support.r* rather than *be-type-of.r* and "is the religious celebration of" to *be-most-important-holiday.r* rather then *be-all-about.r*.

6 Conclusion

This paper introduces an embedding model for question answering with orthogonality regularizer. We show that orthogonality helps to capture the differences between entities and relations and that it helps improve performance on an existing dataset.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive feedback.

References

J. Berant and P. Liang. 2014. Semantic parsing via paraphrasing. In *Annual Meeting for the Association for Computational Linguistics (ACL)*.

J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014a. Question answering with subgraph embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014b. Open question answering with weakly supervised embedding models. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *The 53rd Annual Meeting of the Association for Computational Linguistics and The 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*. ACL.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, Edinburgh, Scotland, UK, July 27-31.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 1156–1165.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2014. Retrofitting word vectors to semantic lexicons. *CoRR*, abs/1411.4166.

Rémi Lebret and Ronan Collobert. 2015. Rehabilitation of count-based models for word vector representations. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part I*, pages 417–429.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Magnus Sahlgren. 2005. An introduction to random indexing. In *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*.

Enpeng Yao, Guoqing Zheng, Ou Jin, Shenghua Bao, Kailong Chen, Zhong Su, and Yong Yu. 2014. Probabilistic text modeling with orthogonalized topics. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 907–910.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 643–648.

Fabio Massimo Zanzotto and Lorenzo Dell'Arciprete. 2012. In *International Conference on Machine Learning (ICML)*.