English-Chinese Knowledge Base Translation with Neural Network

Xiaocheng Feng, Duyu Tang, Bing Qin, Ting Liu
Harbin Institute of Technology, Harbin, China
$\{xcfeng, dytang, qinb, tliu\}$@ir.hit.edu.cn

Abstract

Knowledge base (KB) such as Freebase plays an important role for many natural language processing tasks. English knowledge base is obviously larger and of higher quality than low resource language like Chinese. To expand Chinese KB by leveraging English KB resources, an effective way is to translate English KB (source) into Chinese (target). In this direction, two major challenges are to model triple semantics and to build a robust KB translator. We address these challenges by presenting a neural network approach, which learns continuous triple representation with a gated neural network. Accordingly, source triples and target triples are mapped in the same semantic vector space. We build a new dataset for English-Chinese KB translation from Freebase, and compare with several baselines on it. Experimental results show that the proposed method improves translation accuracy compared with baseline methods. We show that adaptive composition model improves standard solution such as neural tensor network in terms of translation accuracy.

1 Introduction

Knowledge base (KB) like Freebase[1] and Yago[2] has attracted a lot of attention in both research and industry communities. Knowledge Base contains massive triples (entries), each of which is a fact consisting of two arguments and one predicate, such as (*Una White, profession, Nurse*). A large, high-quality KB is valuable and can be applied to many natural language processing and information retrieval tasks (Graupmann et al., 2005; Hotho et al., 2006; Ferrández et al., 2009; Bouma et al., 2009; Shi et al., 2016; Feng et al., 2016). Let us take Freebase as an example, English KB contains 2.7 billion entries and the accuracy is higher than 80%. This is obviously larger and better than a KB with less amount of entries such as Chinese.

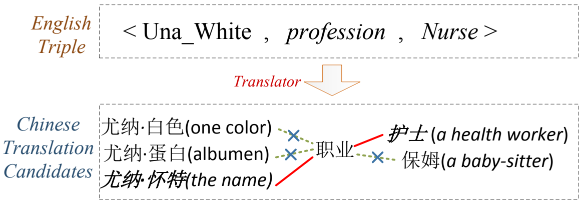

Figure 1: An example of translation ambiguity in English-Chinese KB translation.

A straightforward way to enrich Chinese KB is to directly translate English KB (source) to Chinese (target) based on the surface texts of a triple with existing machine translation system. However, we find that they suffer from the problem of ambiguity. An example is given in Figure 1. The argument "*nurse*" has two translation candidates namely "护士" (a person taking care of sick people in hospital) and "保姆" (baby sitter). "*Una White*" have three translation candidates, so that there are totally six ambiguous

[1] http://en.wikipedia.org/wiki/Freebase

[2] http://en.wikipedia.org/wiki/YAGO_(database)

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 2935–2944, Osaka, Japan, December 11-17 2016.

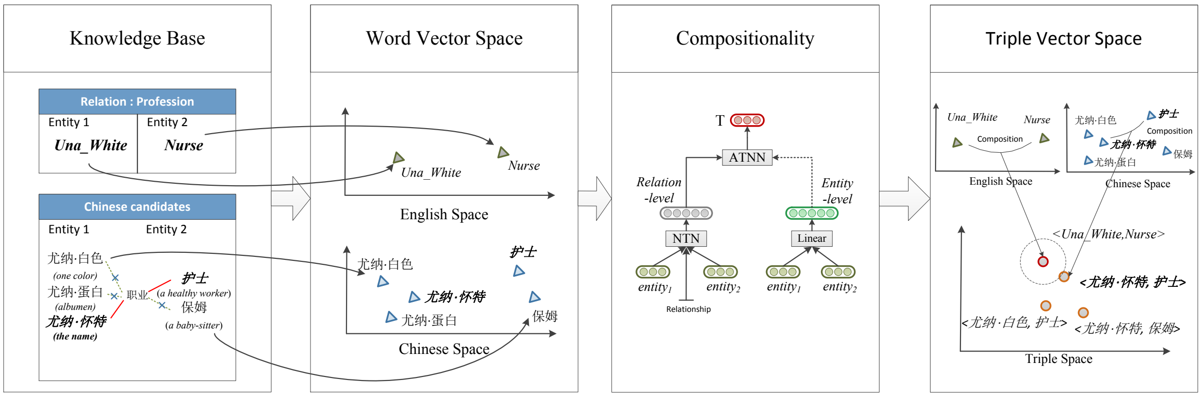

Figure 2: An illustration of the neural network approach for English-Chinese KB translation. It corresponds to the translation example as given in Figure 1.

candidates according to Cartesian product. A preliminary statistical analysis shows that more than half of Freebase translations (English→Chinese) are ambiguous.

There are two main challenges to effectively disambiguate these translated triples. The first challenge is how to effectively model the semantics of English triple and Chinese triple in a unified space. It is preferable to learn a projection function, which maps both English triples and Chinese triples in the same semantic space. The second challenge is how to build a robust KB translator without labor-intensive feature engineering.

In this paper, we address these two challenges by presenting an adaptive neural network to translate English KB into Chinese. Given an English triple and a list of translated Chinese triple candidates, the method assigns a scalar to each translation pair to represent their semantic relatedness. Specifically, we represent each KB triple from the embeddings of words it contains, and introduce an adaptive composition model to effectively capture the semantic composition between arguments of a triple. Compared to previous triple composition methods, the adaptive policy is inspired by highway network (Srivastava et al., 2015b), which is a dynamic calculating process based on different triples rather than a fixed model. In this way, source triple and target triple are naturally encoded in the same semantic vector space. We design a ranking-type hinge loss function to effectively train the parameters of neural networks.

We evaluate the effectiveness of our method on a manually created corpus. We conduct experiments in two settings. Empirical results show that the proposed method consistently outperforms baseline methods. We also show that the use of gated neural network improves strong composition models such as neural tensor network (Socher et al., 2013b) in terms of translation accuracy. The main contributions of this work are as follows:

- We introduce an approach based on representation learning for English-Chinese KB translation in this paper.
- We present a gated neural network to adaptively integrate entity and relational level evidences in triple representation.
- We build a dataset for English-Chinese KB translation, and report the superior performance of our method over baseline methods on it.

2 The Approach

In this section, we present our neural network method for KB Translation in detail. Figure 2 displays a high-level overview of the approach. Given an English triple as input, we first get the candidate Chinese triples (Section 2.1). Afterwards, the semantic representations of English triples and Chinese triples are modeled with neural network (Section 2.2), which are further used for triple ranking (Section 2.3).

2.1 Candidate Generation

In general, a triple in KB is composed of two entities and a relation. In this work, we use English triples from Freebase as the input source, which contains a small number of pre-defined relations. Therefore, we only translate the entities of a triple, while regarding the relation as given. To this end, it is intuitive to use existing text translator to directly translate entities for obtaining candidates. We try to feed English entities to Bing translator, however, a large portion of them can not be translated directly. This stems from the fact that majority of entities are names, proper nouns and named entities which are not well covered in existing translator. Therefore, we use a heuristic method to handle the entities not covered by Bing translator. We split an entity as individual words, and compose the translation results of words as its translation candidate. For example, "*Una White*" in Figure 2.

2.2 Semantic Composition for KB Triple

This section introduces a neural network approach to learn continuous representation for English and Chinese KB triple. We extend this principle in this paper and state that the meaning of a triple is composed from the meanings of entities, relations as well as their correlations.

We find that directly translating entity literally is not effective enough for English-Chinese triple translation. Let us take (Una White, Profession, Nurse) in Figure 1 as an example. The argument "*nurse*" has two translation candidates namely "护士" (a person taking care of sick people in hospital) and "保姆" (baby sitter). "*Una White*" has three translation candidates, so that there are totally six ambiguous candidates according to Cartesian product. To effectively handle this ambiguity problem, we develop an adaptive neural network approach to produce triple representations by effectively capturing entity semantics and the relations between them.

We first describe entity and relational representation. After that, an adaptive composition model is introduced to produce triple representations by automatically combining entity and relational semantics.

2.2.1 Entity Representation

We describe the method to learn continuous representation for each entity. Since an entity is typically a phrase consisting of 2 or 3 words, we average the continuous word representation as the entity representation (Socher et al., 2013b).

Formally, we represent each word as a distributed, continuous and dense vector, which is also known as word embedding (Bengio et al., 2003; Mikolov et al., 2013). These word vectors are stacked in an embedding matrix $L \in \mathbb{R}^{d \times |V|}$, where $|V|$ is word vocabulary size and d is the dimension of each word vector. These word vectors can be randomly initialized from a uniform distribution $U(-0.001, 0.001)$, regarded as parameters of neural networks, and jointly learned with task-specific objectives. Alternatively, the embedding of a word can be trained based on its context information in large-scale text corpus. We use the latter approach since it can make better use of the semantics of words. We learn word embeddings with word2vec[3], which is one of state-of-the-art embedding learning algorithms and widely used for many natural language processing tasks. We learn English word embedding from Wikipedia dump[4], and learn Chinese word embedding from Baike texts[5].

To compose the entity representation from the embedding of words it contains, we follow Socher et al. (2013b) and average the continuous word representation as entity representation. Recursive Neural Network is not suitable to represent entity because entities are typically people names which do not contain explicit compositional structure.

Since the English word vectors and Chinese word vectors are separately trained without using bilingual parallel corpus, these word vectors are mapped into different semantic spaces. This is not desirable for comparing the semantic relatedness between English triple and Chinese triple. We use linear layers to transform English and Chinese word vectors in a same semantic vector space. A simple linear layer is calculated as $v_e = We + b$, where W and b are the parameters. One could also learn bilingual

[3] code.google.com/p/word2vec/

[4] https://dumps.wikimedia.org/

[5] http://baike.baidu.com/

word vectors simultaneously from bilingual parallel corpus with tailored learning algorithm (Zou et al., 2013). We leave this as a future work and we believe our method could benefit from the bilingual word embeddings.

2.2.2 Relational Representation

We model the semantic relatedness between entities in this part. The basic idea is that the semantic relatedness between entities is determined by the semantics of entities and their relations. Based on this, we utilize neural tensor network, which is one of state-of-the-art semantic composition approach for natural language processing tasks (Mitchell and Lapata, 2010; Socher et al., 2013a; Jenatton et al., 2012).

A standard neural tensor with rank 3 is essentially a list of bilinear neural layers, each of which takes two vectors as inputs and outputs a real-valued scalar with element wise multiplication. Furthermore, relation-specific neural tensor can be exploited to make better use of the relation between entities, which is calculated as follows.

$$v_r = e_1^T W_R^{[1:k]} e_2 \tag{1}$$

where k is the length of output vector, e_1 and e_2 are the d-dimensional embeddings of two entities in a given triple, $W_R^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ stands for the parameters of tensor. Each element in v_r is computed by one slice $i \in \{1, ..., k\}$ of tensor.

2.2.3 Adaptive Neural Network

We have previously obtained entity representation and relational representation, both of which play important roles for representing the meaning of a triple. Furthermore, a better approach should benefit from both aspects, and integrate them in triple semantic with an automatic method. To this end, we introduce a gated neural network in this part. It takes entity and relational vectors of a triple as input, and adaptively produces the composed continuous representation of them.

Given v_e and v_r as inputs, a traditional compositional function is to concatenate v_e and v_r and feed them to a linear layer (Socher et al., 2011), which is calculated as Equation 2. Despite its computational efficiency, tied parameters cannot easily capture the complex linguistic phenomena in natural language expressions.

$$\tilde{v} = tanh(W_e v_e + W_r v_r + b) \tag{2}$$

$$\alpha = \sigma(W_{eg} v_e + W_{rg} v_r + b_g) \tag{3}$$

$$v(t) = \alpha \cdot v_r + (1 - \alpha) \cdot \tilde{v} \tag{4}$$

Therefore, we add a neural gate to change parameter values for different input vectors v_e and v_r, which is partly inspired by the recent success of gated recurrent neural network (Cho et al., 2014; Chung et al., 2015) and Long Short-Term Memory (Hochreiter and Schmidhuber, 1997; Tai et al., 2015). And our gated neural network is inspired by highway network, which allow the model to suffer less from the vanishing gradient problem (Srivastava et al., 2015a; Srivastava et al., 2015b). The gate takes v_e and v_r as inputs, and outputs as a weight $\alpha \in [0, 1]$, which linearly weights the two parts. Specifically, the gate is calculated as Equation 3, where σ is standard sigmoid function, W_{eg}, W_{rg} and b_g are parameters. Triple representation $v(t)$ is calculated as given in Equation 4, which linearly weights the candidate composed representation $\tilde{v}$ and relational representation v_r. In this way, entity representation and relational representation are adaptively encoded in the semantic representation of a triple.

2.3 English-Chinese Triple Translation

Given an English triple t_e and a list of candidate Chinese triples $\{t_{c1}, ...t_{cj}, ... \ t_{ck}\}$, we select the most relevant Chinese candidate in terms of semantic as the translation answer. To this end, we need to formalize a scoring function $f(t_e, t_{cj})$, which is capable of measuring the semantic relatedness between an English triple t_e and a Chinese candidate triple t_{cj}. Specifically, we apply the continuous triple vector learned in Section 2.2.3 as English and Chinese triple representations without any feature engineering. We use standard L1 and L2 norms as the dissimilarity measure f, namely:

$$f(t_e, t_{cj}) = ||v(t_e) - v(t_{cj})||_p \tag{5}$$

where $p = 1$ means L1 norm, and $p = 2$ stands for L2 norm.

To effectively estimate the parameters of the neural networks, we use a ranking type loss function based on the intuition of noise contrastive estimation (Gutmann and Hyvärinen, 2010), which has been exploited as an effective training objective in deep learning community (Socher et al., 2013b).

In this paper, the basic idea of the optimizing objective is that: the distance between an English triple and its correct translation candidate $f(t_e, t_c)$ should get a lower score than the distance between the source triple and a *corrupted* incorrect candidate triple $f(t_e, t_c')$ by a margin of 1.

$$Loss = \sum_{i=1}^{N} max(0, 1 - f(t_e, t_c') + f(t_e, t_c)) \tag{6}$$

where N is the number of training instances. Given a correct Chinese triple (e_1, r, e_2), we generate two corrupted triples by randomly replacing one entity at one time, resulting in (e_1, r, e_2') and (e_1', r, e_2). We train the neural networks with supervised learning using stochastic gradient descent. We take the derivative of the loss regarding the parameters with standard back-propagation. We learn 50-dimensional English and Chinese word embeddings using word2vec with default setting. The vocabulary size of English and Chinese word embeddings are 32K and 30K, respectively. For each relation type, we use relation untied parameters, randomly initialize the values of $W_e, W_r, W_{eg}, W_{rg}, W_R^1, W_R^2, u'$, b and b_g from a uniform distribution $U(-0.01/L, 0.01/L)$ where L is the input length of a neural layer, set the hidden length as 30, set the learning rate as 0.03 and tune the training round on development set.

3 Experiment

We apply the proposed method for English-Chinese KB translation to evaluate its effectiveness. We describe the data statistics, experimental settings and empirical results.

3.1 Experiment Settings

For the task of English-Chinese KB translation, since there is no publicly available benchmark dataset, we manually annotate a dataset by ourselves. We use Freebase as the English source, which is widely used in the field of KB population and KB completion. There are several relation types in Freebase, we only select "Profession" and "Cause of Death" in this work because they are representative relation types with large number of instances. We leave other relation types as future work. We use Bing translator to get the translation candidates, and employ two experts to annotate the best result among candidate list. The disagreements during annotation are fixed by detailed discussion.We randomly split the dataset as training, development and testing sets. The statistical information of the datasets are given in Table 1.

Relation Type	#Train	# Dev	# Test
Profession	3,000	500	500
Cause of death	2,000	500	500

Table 1: The statistics for Freebase including two different relations.

We conduct experiments in a supervised learning framework. For each relation type, we train the model on training set, tune parameters on dev set and evaluate on test set. We use P@1 (Manning and Schütze, 1999) as the evaluation metric, which indicates whether the first ranked translation results is the correct answer.

- Surface Matching. Given an English triple (e_1, r, e_2), we first get the translation candidates of e_1 and e_2. After that, we select the top-ranked entity in each set, and merge them as the best translate result.

Model	L1 Norm			L2 Norm		
	Profession	C-death	Avg	Profession	C-death	Avg
Surface Matching	52.4	51.6	52.0	52.4	51.6	52.0
Hints Similarity	58.2	56.4	57.3	58.2	56.4	57.3
Entity Model	70.4	64.6	67.5	72.2	67.4	69.9
Relational (Tensor) Model	71.8	65.2	68.5	72.4	68.6	70.5
Relational (Tensor*) Model	72.4	67.2	69.8	72.8	69.8	71.3
Average (Entity + Relational)	72.8	67.6	70.2	73.0	70.2	71.6
Linear (Entity + Relational)	73.2	68.0	70.6	73.6	70.6	72.1
Full Model	75.0	70.4	72.7	75.6	71.8	73.7

Table 2: Comparison of accuracy of the different models for English-Chinese KB translation. We run experiments in L1 Norm and L2 Norm. Evaluation metric is P@1.

- Hints Similarity. After obtaining the list of candidates with Cartesian product, we measure the similarity between entities in a candidate triple with web search. We concatenate two entities as a query and put them in a Chinese search engine. We count the co-occurrence frequency in snippets, and select the top ranked candidate as the answer.

Our model has several variations, which are detailed as below.

- Entity Model. We represent a triple by only using entity representation, without leveraging relational representation (Bordes et al., 2012) .
- Relational Model. We represent a triple by only using relational representation, without using entity representation. In Relational (Tensor) Model, the neural calculator only uses multiplicative composition function as described in Section 2.2.2. In Relational (Tensor*) Model, the neural calculator includes an additional linear layer, which is exploited in Socher et al. (2013b) and calculated as below.

$$v_t = [e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R] \tag{7}$$

- Average (Entity + Relational). In this setting, we represent a triple by averaging its entity vector and relational vector, without using gated neural network.

$$v_t = \frac{v_e + v_r}{2} \tag{8}$$

- Linear (Entity + Relational). In this setting, we represent a triple by concatenating its entity vector and relational vector, and composing them with standard linear layer. This is a special case of the gated neural network, where α is always set to zero.

3.2 Results and Analysis

Table 2 shows the empirical results of the baseline methods and our method on two relations. We can find that the performances of these methods are consistent on two relations. Among all these algorithms, *Surface Matching* is the worst performer. The reason lies in that it does not capture the interaction between two entities in a triple, which is very important for KB translation. *Hints Similarity* outperforms *Surface Matching* by taking into account the relatedness of entities by web search results. However, its improvement over *Surface Matching* is not significant enough because it ignores the relation type. That is to say, the semantic similarities between two entities in a triple remain the same for different relation types (e.g. Profession, Cause of Death). This is problematic as the relation plays an important role in discovering entity similarity. Let us take “Barack Obama （奥巴马）” and “Honolulu（火奴鲁鲁）” as an example. Their similarity in terms of “born in” relation is high, but the similarity regarding “working place” should be extremely lower.

Table 2 shows that neural network models outperform *Surface Matching* and *Hints Similarity* method, which shows the powerfulness of neural network based representation learning methods. This is because that neural network methods map English triples and Chinese triples into a unified semantic vector space, which shares some characteristics with the human who is assigned to do this task.

Among all these neural network methods, *Entity Model* is the worst method because it captures the semantics of entities separately while ignoring the semantic interaction between them. On the other hand, *Relational (Tensor) Model* only use the relational information of a triple. We can find that it shows slight improvements over *Entity Model*, which indicates the importance of relation of triple for KB translation.

Relational (Tensor) Model* is an enhanced *Relational (Tensor) Model*, and can also be viewed as a tensor composition function added by a standard linear composition function (Socher et al., 2013b). We can find that Relational (Tensor*) Model yields better performances than previous two neural models. The full model yields the best performances among all baseline methods by simultaneously leveraging entity representation, relational representation and their interaction in an adaptive method.

Average (Entity + Relational) is a straight-forward method to compose entity- and relational- representations. From Table 2, we can find that *Average* does not yield obvious improvements over previous neural models. The main reason is that average function fails to model the interaction between entity vector and relational vector, which is important to effectively capture the complex linguistic phenomena in KB triple. In *Linear (Entity + Relational)*, we concatenate v_e and v_r, and calculate their semantic composition with a simple linear layer (Socher et al., 2011). It shows some improvements over *Average (Entity + Relational)*, which demonstrates the importance of semantic composition algorithm for obtaining semantic representation of a triple.

3.3 The Effect of Gated Neural Network

We explore the effectiveness of the gated neural model for English-Chinese KB translation on our datasets. We set the gated α as a fixed value from 0.1 to 1.0, increased by 0.1. This corresponds to a special case of gated neural network with a fixed weighted ratio between v_r and $\tilde{v}$. The model with α = 1.0 means that the representation of a triple only comes from relational representation, without using any entity representation as mentioned in Section 2.2.1.

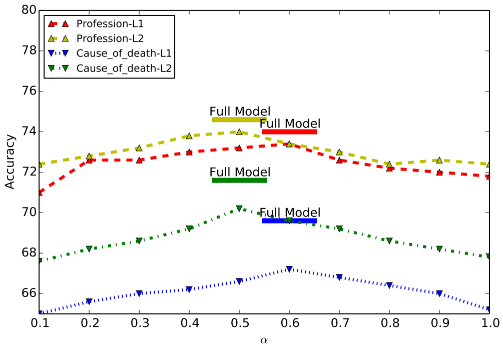

Figure 3: Experimental results with different $\alpha 1$ on the datasets.

We run experiments on two relation types with L1 and L2 norms, respectively. The results are illustrated in Figure 3. We can see that the performance of our full model is obviously better than the model with fixed trade-off weights. This is partly because that it is hard to measure the importances of entity representation and relational representation with a fixed weight for the complex phenomena in KB. The results also reveal the importance of an adaptive weighting strategy for semantic composition.

4 Related Work

We briefly describe existing studies on representation learning for natural language processing, learning continuous triple representation and gated recurrent neural network.

It is well accepted that feature representation is extremely important for natural language processing tasks. The main reason is that the effectiveness of a machine learner is highly dependent on the choice of data representation (Bengio et al., 2013). In past few decades, many studies leverage human ingenuity and prior knowledge to design hand-crafted features. Despite the effectiveness of feature engineering in some tasks, it is time consuming and typically fails to extract the discriminative information from the data. Recently, neural networks show their strengths in learning continuous representations of word/phrase (Mikolov et al., 2013), sentence (Socher et al., 2013c), document (Le and Mikolov, 2014), KB triple (Bordes et al., 2013) from data without any feature engineering. This study belongs to the family of neural network based representation learning for natural language processing tasks.

There are several neural network approaches proposed to model relational data, especially in the multi-relational case, where different kinds of relations are used to connect various data entities. Previous works focus on knowledge link prediction and triplet classification. Bordes et al. (2011) provide a structured embedding model where the regression loss was replaced by a ranking loss for learning embeddings of entities. Bordes et al. (2012) introduced a semantic matching energy function to map different instances in the same semantic vector space. Bordes et al. (2013) exploited a canonical model, which modeled relations by regarding the task as a translations operation on the low-dimensional embeddings of entities. Wang et al. (2014) made an extension on the translation model of TransE (Bordes et al., 2013) by projecting KB triple in relation-specific hyperplane. In this way, they can preserve the mapping properties of relation to some extent. Lin et al. (2015) considered that an entity may have multiple aspects, and different relations focused on different aspects of entities. They introduced TransR by representing entities and relations in distinct semantic vector space. Another related approach is introduced by Socher et al. (2013a), which used a neural tensor network to learn relational compositionality. Our relational representation method is similar to Socher et al. (2013a), which is on the basis of multiplicative vector-based semantic composition (Mitchell and Lapata, 2010).

The use of gated neural network in this paper shares some characteristics with the emerging gated recurrent neural network (Cho et al., 2014; Chung et al., 2015) and Long Short-Term Memory (Hochreiter and Schmidhuber, 1997; Tai et al., 2015). Standard recurrent neural network uses a set of shared parameters to represent a sequence of variable length to a vector representation. It suffers from the problem of vanishing gradient, which means that the influence of a given input on the hidden layer either decays or blows up exponentially. LSTM and gated recurrent neural network address this problem by adding several neural gates (e.g. input and forget gates) to adaptively memorize new content and forget history content. The gated neural network used in this work aims at integrating entity representation and relational representation in triple vector in an adaptive way.

5 Conclusion

We introduce a neural network approach for Knowledge Base (KB) translation from English (source) to Chinese (target) in this paper. We represent a triple in KB with an adaptive composition model, which produces triple representation by capturing entity- and relational- level information. The parameters of neural networks are effectively estimated with a ranking-type hinge loss function. We compare against several baseline methods on two KB translation datasets. Experimental results show that, our method performs better than baseline methods. In addition, we show that the newly introduced adaptive composition model improves standard composition method such as neural tensor network in terms of translation accuracy.

6 Acknowledgments

The authors give great thanks to Lifu Huang (RPI) and Shen Liu (HIT) for the fruitful discussions and related web crawlers work. We also would like to thank four anonymous reviewers for their valuable comments and suggestions. This work was supported by the National High Technology Development 863 Program of China (No. 2015AA015407), National Natural Science Foundation of China (No. 61632011 and No. 61370164).

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Analysis and Machine Intelligence*.

Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, pages 127–135.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Gosse Bouma, Sergio Duarte, and Zahurul Islam. 2009. Cross-lingual alignment and completion of wikipedia templates. In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies*, pages 21–29. ACL.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. *arXiv preprint arXiv:1502.02367*.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Bing Qin, Heng Ji, and Ting Liu. 2016. A language-independent neural network for event detection. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 66.

Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz. 2009. Exploiting wikipedia and eurowordnet to solve cross-lingual question answering. *Information Sciences*, 179(20):3473–3488.

Jens Graupmann, Ralf Schenkel, and Gerhard Weikum. 2005. The spheresearch engine for unified ranked retrieval of heterogeneous xml and web documents. In *Proceedings of the 31st international conference on Very large data bases*, pages 529–540. VLDB Endowment.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pages 297–304.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. 2006. *Information retrieval in folksonomies: Search and ranking*. Springer.

Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *ICML*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *NIPS*.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. 2016. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Richard Socher, J. Pennington, E.H. Huang, A.Y. Ng, and C.D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP 2011*.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing with compositional vector grammars. In *ACLAnnual Meeting of the Association for Computational Linguisticsm*.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Y Ng. 2013b. Reasoning with neural tensor networks for knowledge base completion. *NIPS*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013c. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP 2013*, pages 1631–1642.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015a. Training very deep networks. In *NIPS 2015*, pages 2368–2376.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015b. Highway networks. *arXiv preprint arXiv:1505.00387*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Zhen Wang, 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*, pages 1112–1119.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.