Connecting Language and Knowledge with Heterogeneous Representations for Neural Relation Extraction

Peng Xu
Department of Computing Science
University of Alberta
Edmonton, Canada
pxu4@ualberta.ca

Denilson Barbosa
Department of Computing Science
University of Alberta
Edmonton, Canada
denilson@ualberta.ca

Abstract

Knowledge Bases (KBs) require constant updating to reflect changes to the world they represent. For general purpose KBs, this is often done through Relation Extraction (RE), the task of predicting KB relations expressed in text mentioning entities known to the KB. One way to improve RE is to use KB Embeddings (KBE) for link prediction. However, despite clear connections between RE and KBE, little has been done toward properly unifying these models systematically. We help close the gap with a framework that unifies the learning of RE and KBE models leading to significant improvements over the state-of-the-art in RE. The code is available at `https://github.com/billy-inn/HRERE`.

1 Introduction

Knowledge Bases (KBs) contain structured information about the world and are used in support of many natural language processing applications such as semantic search and question answering. Building KBs is a never-ending challenge because, as the world changes, new knowledge needs to be harvested while old knowledge needs to be revised. This motivates the work on the Relation Extraction (RE) task, whose goal is to assign a KB relation to a *phrase* connecting a pair of entities, which in turn can be used for updating the KB. The state-of-the-art in RE builds on neural models using distant (a.k.a. weak) supervision (Mintz et al., 2009) on large-scale corpora for training.

A task related to RE is that of Knowledge Base Embedding (KBE), which is concerned with representing KB entities and relations in a vector space for predicting missing links in the graph. Aiming to leverage the similarities between these tasks, Weston et al. (2013) were the first to show that *combining* predictions from RE and KBE models was beneficial for RE. However, the way in which they combine RE and KBE predictions is rather naive (namely, by adding those scores). To the best of our knowledge, there have been no systematic attempts to further unify RE and KBE, particularly during model *training*.

We seek to close this gap with HRERE (Heterogeneous REpresentations for neural Relation Extraction), a novel neural RE framework that learns language and knowledge representations *jointly*. Figure 1 gives an overview. HRERE's backbone is a bi-directional long short term memory (LSTM) network with multiple levels of attention to learn representations of text expressing relations. The knowledge representation machinery, borrowed from **ComplEx** (Trouillon et al., 2016), nudges the language model to agree with facts in the KB. Joint learning is guided by three loss functions: one for the language representation, another for the knowledge representation, and a third one to ensure these representations do not diverge. In effect, this contributes to HRERE's generalization power by preventing over-fitting by either model.

We build on state-of-the-art methods for learning the separate RE and KBE representations and on learning tools that allow us to scale to a moderately large training corpus. (We use a subset of Freebase with 3M entities as our KB.) We validate our approach on an established benchmark against state-of-the-art methods for RE, observing not only that our base model significantly outperforms previous methods, but also the fact that jointly learning the heterogeneous representations consistently brings in improvements. To the best of our knowledge, ours is the first principled framework to combine and jointly learn heterogeneous representations from both language and knowledge for the RE task.

Contributions. This paper describes and evaluates a novel neural framework for jointly learning

Proceedings of NAACL-HLT 2019, pages 3201–3206
Minneapolis, Minnesota, June 2 - June 7, 2019.

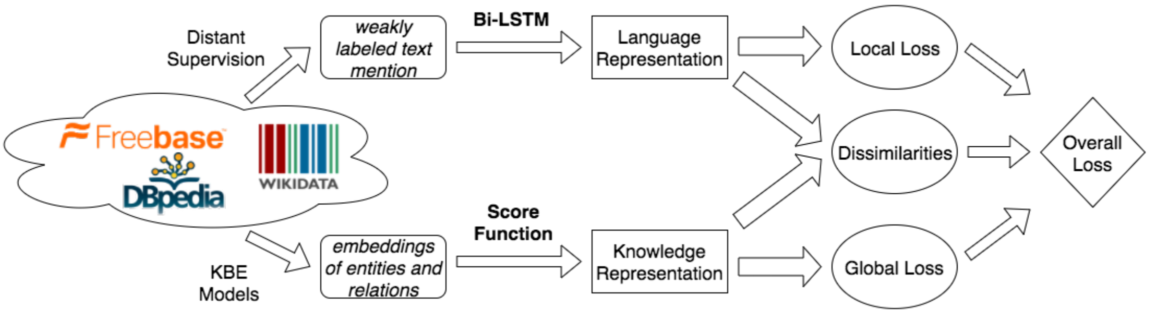

Figure 1: Workflow of the proposed framework.

representations for RE and KBE tasks that uses a cross-entropy loss function to ensure both representations are learned together, resulting in significant improvements over the current state-of-the-art for the RE task.

2 Related Work

Recent neural models have been shown superior to approaches using hand-crafted features for the RE task. Among the pioneers, Zeng et al. (2015) proposed a piecewise convolutional network with multi-instance learning to handle weakly labeled text mentions. Recurrent neural networks (RNN) are another popular architecture (Wu et al., 2017). Similar fast progress has been seen for the KBE task for representing entities and relations in KBs with vectors or matrices. Bordes et al. (2013) introduced the influential translation-based embeddings (TransE), while Yang et al. (2014) leveraged latent matrix factorization in their DistMult method. We build on ComplEx (Trouillon et al., 2016), which extends DistMult into the complex space and has been shown significantly better on several benchmarks.

Weston et al. (2013) were the first to connect RE and KBE models for the RE task. Their simple idea was to train the two models *independently* and only combine them at inference time. While they showed that combining the two models is better than using the RE model alone, newer and better models since then have obviated the net gains of such a simple strategy (Xu and Barbosa, 2018). We propose a much tighter integration of RE and KBE models: we not only use them for prediction, but also *train* them together, thus mutually reinforcing one another.

Recently, many methods have been proposed to use information from KBs to facilitate relation extraction. Sorokin and Gurevych (2017) considered other relations in the sentential context while predicting the target relation. Vashishth et al. (2018) utilized additional side information from KBs for improved RE. However, these methods didn't leverage KBE method to unify RE and KBE in a principled way. Han et al. (2018) used a mutual attention between KBs and text to perform better on both RE and KBE, but their method was still based on TransE (Bordes et al., 2013) which can not fully exploit the advantage of the information from KBs.

3 Background and Problem

The goal in the task of Relation Extraction is to predict a KB relation that holds for a pair of entities given a set of sentences mentioning them (or *NA* if no such relation exists). The input is a KB Ψ with relation set $\mathcal{R}_\Psi$, a set of relations of interest $\mathcal{R}$, $\mathcal{R} \subseteq \mathcal{R}_\Psi$, and an automatically labelled training dataset $\mathcal{D}$ obtained via distant supervision. Given a sentence mentioning entities h, t, the output is a relation $r \in \mathcal{R}$ that holds for h, t or the catch-all relation *NA* if no such r exists.

Knowledge Base and Distant Supervision. As customary, we denote a KB Ψ with relation scheme $\mathcal{R}_\Psi$ as a set of *triples* $\mathcal{T}_\Psi = \{(h, r, t) \in \mathcal{E}_\Psi \times \mathcal{R}_\Psi \times \mathcal{E}_\Psi\}$, where $\mathcal{E}_\Psi$ is the set of entities of interest. Distant supervision exploits the KB to automatically annotate sentences in a corpus containing mentions of entities with the relations they participate in. Formally, a labeled dataset for relation extraction consists of fact triples $\{(h_i, r_i, t_i)\}_{i=1}^N$ and a multi-set of extracted sentences for each triple $\{\mathcal{S}_i\}_{i=1}^N$, such that each sentence $s \in \mathcal{S}_i$ mentions both the head entity h_i and the tail entity t_i.

Problem Statement. Given an entity pair (h, t) and a set of sentences $\mathcal{S}$ mentioning them, the RE task is to estimate the probability of each relation in $\mathcal{R} \cup \{NA\}$. Formally, for each relation r, we want to predict $P(r \mid h, t, \mathcal{S})$.

In practice, the input set of sentences $\mathcal{S}$ can have arbitrary size. For the sake of computational efficiency, we normalize the set size to a fixed number T by splitting large sets and oversampling small ones. We also restrict the length of each sentence in the set by a constant L by truncating long sentences and padding short ones.

4 Methodology

We now go over the details of our framework outlined in Figure 1 for unifying the learning of the language and the knowledge representations used for relation extraction. In a nutshell, we use LSTM with attention mechanisms for language representation and we follow the approach of Trouillon et al. (2016) for KB embedding.

4.1 Language Representation

Input Representation. For each word token, we use pretrained word embeddings and randomly initialized position embeddings (Zeng et al., 2014) to project it into $(d_w + d_p)$-dimensional space, where d_w is the size of word embedding and d_p is the size of position embedding.

Sentence Encoder. For each sentence s_i, we apply a non-linear transformation to the vector representation of s_i to derive a feature vector $z_i = f(s_i; \theta)$ given a set of parameters θ. In this paper, we adopt bidirectional LSTM with d_s hidden units as $f(s_i; \theta)$ (Zhou et al., 2016).

Multi-level Attention Mechanisms. We employ attention mechanisms at both word-level and sentence-level to allow the model to softly select the most informative words and sentences during training (Zhou et al., 2016; Lin et al., 2016). With the learned language representation $\mathbf{s}_L$, the conditional probability $p(r|\mathcal{S}; \Theta^{(L)})$ is computed through a *softmax* layer, where $\Theta^{(L)}$ is the parameters of the model to learn language representation.

4.2 Knowledge Representation

Following the score function ϕ and training procedure of Trouillon et al. (2016), we can get the knowledge representations $e_h, w_r, e_t \in \mathbb{C}^{d_k}$. With the knowledge representations and the scoring function, we can obtain the conditional probability $p(r|(h,t); \Theta^{(G)})$ for each relation r:

$$p(r|(h,t); \Theta^{(G)}) = \frac{e^{\phi(e_h, w_r, e_t)}}{\sum_{r' \in \mathcal{R} \cup \{NA\}} e^{\phi(e_h, w_{r'}, e_t)}}$$

where $\Theta^{(G)}$ corresponds to the knowledge representations $e_h, w_r, e_t \in \mathbb{C}^{d_k}$. Since $NA \notin \mathcal{R}_\Psi$, we use a randomized complex vector as w_{NA}.

4.3 Connecting the Pieces

As stated, this paper seeks an elegant way of connecting language and knowledge representations for the RE task. In order to achieve that, we use separate loss functions (recall Figure 1) to guide the language and knowledge representation learning and a third loss function that ties the predictions of these models thus nudging the parameters towards agreement.

The cross-entropy losses based on the language and knowledge representations are defined as:

$$\mathcal{J}_L = -\frac{1}{N} \sum_{i=1}^{N} \log p(r_i | \mathcal{S}_i; \Theta^{(L)}) \quad (1)$$

$$\mathcal{J}_G = -\frac{1}{N} \sum_{i=1}^{N} \log p(r_i | (h_i, t_i); \Theta^{(G)}) \quad (2)$$

where N denotes the size of the training set. Finally, we use a cross-entropy loss to measure the dissimilarity between two distributions, thus connecting them, and formulate model learning as minimizing $\mathcal{J}_D$:

$$\mathcal{J}_D = -\frac{1}{N} \sum_{i=1}^{N} \log p(r_i^* | \mathcal{S}_i; \Theta^{(L)}) \quad (3)$$

where $r_i^* = \arg\max_{r \in \mathcal{R} \cup \{NA\}} p(r|(h_i, t_i); \Theta^{(G)})$.

4.4 Model Learning

Based on Eq. 1, 2, 3, we form the joint optimization problem for model parameters as

$$\min_{\Theta} \mathcal{J} = \mathcal{J}_L + \mathcal{J}_G + \mathcal{J}_D + \lambda \|\Theta\|_2^2 \quad (4)$$

where $\Theta = \Theta^{(L)} \cup \Theta^{(G)}$. The knowledge representations are first trained on the whole KB independently and then used as the initialization for the *joint* learning. We adopt the stochastic gradient descent with mini-batches and Adam (Kingma and Ba, 2014) to update Θ, employing different learning rates lr_1 and lr_2 on $\Theta^{(L)}$ and $\Theta^{(G)}$ respectively

4.5 Relation Inference

In order to get the conditional probability $p(r|(h,t),\mathcal{S};\Theta)$, we use the weighed average to combine the two distribution $p(r|\mathcal{S};\Theta^{(L)})$ and $p(r|(h,t);\Theta^{(G)})$:

$$\begin{aligned} p(r|(h,t),\mathcal{S};\Theta) &= \alpha * p(r|\mathcal{S};\Theta^{(L)}) \\ &+ (1-\alpha) * p(r|(h,t);\Theta^{(G)}). \end{aligned} \tag{5}$$

where α is the combining weight of the weighted average. Then, the predicted relation $\hat{r}$ is

$$\hat{r} = \underset{r \in \mathcal{R} \cup \{NA\}}{\operatorname{argmax}} \; p(r|(h,t),\mathcal{S};\Theta). \tag{6}$$

5 Experiments

Datasets. We evaluate our model on the widely used **NYT** dataset (Riedel et al., 2010) by aligning Freebase relations mentioned in the New York Times Corpus. Articles from years 2005-2006 are used for training while articles from 2007 are used for testing. As our KB, we used a Freebase subset with the 3M entities with highest degree (i.e., participating in most relations). Moreover, to prevent the knowledge representation from memorizing the true relations for entity pairs in the test set, we removed all entity pairs present in the NYT.

Evaluation Protocol: Following previous work (Mintz et al., 2009), we evaluate our model using held-out evaluation which approximately measures the precision without time-consuming manual evaluation. We report both Precision/Recall curves and Precision@N (P@N) in our experiments, ignoring the probability predicted for the *NA* relation. Moreover, to evaluate each sentence in the test set as in previous methods, we append T copies of each sentence into $\mathcal{S}$ for each testing sample.

Word Embeddings: In this paper, we used the freely available 300-dimensional pre-trained word embeddings distributed by Pennington et al. (2014) to help the model generalize to words not appearing in the training set.

Hyperparameter Settings: For hyperparameter tuning, we randonly sampled 10% of the training set as a development set. All the hyperparameters were obtained by evaluating the model on the development set. With the well-tuned hyperparameter setting, we run each model five times on the whole training set and report the average P@N. For Precision/Recall curves, we just select the results from the first run of each model. For training, we set the iteration number over all the training data as 30. Values of the hyperparameters used in the experiments can be found in Table 1.

learning rate on $\Theta^{(L)}$	lr_1	5×10^{-4}
learning rate on $\Theta^{(K)}$	lr_2	1×10^{-5}
size of word position embedding	d_p	25
state size for LSTM layers	d_s	320
input dropout keep probability	p_i	0.9
output dropout keep probability	p_o	0.7
L2 regularization parameter	λ	0.0003
combining weight parameter	α	0.6

Table 1: Hyperparameter setting

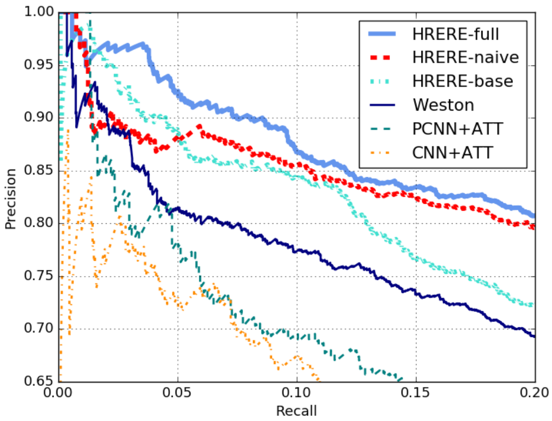

Figure 2: The Precision/Recall curves of previous state-of-the-art methods and our proposed framework.

P@N(%)	10%	30%	50%
Weston	79.3	68.6	60.9
HRERE-base	81.8	70.1	60.7
HRERE-naive	83.6	74.4	65.7
HRERE-full	**86.1**	**76.6**	**68.1**

Table 2: P@N of **Weston** and variants of our proposed framework.

Methods Evaluated. We study three variants of our framework: (1) HRERE-**base**: basic neural model with local loss $\mathcal{J}_L$ only; (2) HRERE-**naive**: neural model with both local loss $\mathcal{J}_L$ and global loss $\mathcal{J}_G$ but without the dissimilarities $\mathcal{J}_D$; (3) HRERE-**full**: neural model with both local and global loss along with their dissimilarities. We compare against two previous state-of-the-art neural models, **CNN+ATT** and **PCNN+ATT** (Lin et al., 2016). We also implement a baseline **Weston** based on the strategy following Weston et al. (2013), namely to combine the scores computed

Relation	Textual Mention	base	naive	full
contains	Much of the **middle east** tension stems from the sense that shiite power is growing, led by **Iran**.	0.311	0.864	**0.884**
place_of_birth	Sometimes I rattle off the names of movie stars from **Omaha**: Fred Astaire, **Henry Fonda**, Nick Nolte . . .	0.109	0.605	**0.646**
country	Spokesmen for **Germany** and Italy in Washington said yesterday that they would reserve comment until the report is formally released at a news conference in **Berlin** today.	0.237	0.200	**0.880**

Table 3: Some examples in NYT corpus and the predicted probabilities of the true relations.

with the methods stated in this paper directly without joint learning.

Analysis. Figure 2 shows the Precision/Recall curves for all the above methods. As one can see, HRERE-**base** significantly outperforms previous state-of-the-art neural models and **Weston** over the entire range of recall. However, HRERE-**base** performs worst compared to all other variants, while HRERE-**full** always performs best as shown in Figure 2 and Table 2. This suggests that introducing knowledge representation consistently results in improvements, which validates our motivating hypothesis. HRERE-**naive** simply optimizes both local and global loss at the same time without attempting to connect them. We can see that HRERE-**full** is not only consistently superior but also more stable than HRERE-**naive** when the recall is less than 0.1. One possible reason for the instability is that the results may be dominated by one of the representations and biased toward it. This suggests that (1) jointly learning the heterogeneous representations bring mutual benefits which are out of reach of previous methods that learn each independently; (2) connecting heterogeneous representations can increase the robustness of the framework.

Case Study. Table 3 shows two examples in the testing data. For each example, we show the relation, the sentence along with entity mentions and the corresponding probabilities predicted by HRERE-**base** and HRERE-**full**. The entity pairs in the sentence are highlighted with bold formatting.

From the table, we have the following observations: (1) The predicted probabilities of three variants of our model in the table match the observations and corroborate our analysis. (2) From the text of the two sentences, we can easily infer that *middle east contains Iran* and *Henry Fonda was born in Omaha*. However, HRERE-**base** fails to detect these relations, suggesting that it is hard for models based on language representations alone to detect implicit relations, which is reasonable to expect. With the help of KBE, the model can effectively identify implicit relations present in the text. (3) It may happen that the relation cannot be inferred by the text as shown in the last example. It's a common wrong labeled case caused by distant supervision. It is a case of an incorrectly labeled instance, a typical occurrence in distant supervision. However, the fact is obviously true in the KBs. As a result, HRERE-**full** gives the underlying relation according to the KBs. This observation may point to one direction of de-noising weakly labeled textual mentions generated by distant supervision.

6 Conclusion

This paper describes an elegant neural framework for jointly learning heterogeneous representations from text and from facts in an existing knowledge base. Contrary to previous work that learn the two disparate representations independently and use simple schemes to integrate predictions from each model, we introduce a novel framework using an elegant loss function that allows the proper connection between the the heterogeneous representations to be learned seamlessly during training. Experimental results demonstrate that the proposed framework outperforms previous strategies to combine heterogeneous representations and the state-of-the-art for the RE task. A closer inspection of our results show that our framework enables both independent models to enhance each other.

Acknowledgments

This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada and a gift from Diffbot Inc.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1779–1784.

Peng Xu and Denilson Barbosa. 2018. Investigations on knowledge base embedding for relation prediction and extraction. *arXiv preprint arXiv:1802.02114*.

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

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Emnlp*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.