Representing Text for Joint Embedding of Text and Knowledge Bases

Kristina Toutanova
Microsoft Research
Redmond, WA, USA

Danqi Chen*
Computer Science Department
Stanford University

Patrick Pantel
Microsoft Research
Redmond, WA, USA

Hoifung Poon
Microsoft Research
Redmond, WA, USA

Pallavi Choudhury
Microsoft Research
Redmond, WA, USA

Michael Gamon
Microsoft Research
Redmond, WA, USA

Abstract

Models that learn to represent textual and knowledge base relations in the same continuous latent space are able to perform joint inferences among the two kinds of relations and obtain high accuracy on knowledge base completion (Riedel et al., 2013). In this paper we propose a model that captures the compositional structure of textual relations, and jointly optimizes entity, knowledge base, and textual relation representations. The proposed model significantly improves performance over a model that does not share parameters among textual relations with common sub-structure.

1 Introduction

Representing information about real-world entities and their relations in structured knowledge base (KB) form enables numerous applications. Large, collaboratively created knowledge bases have recently become available e.g., Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007), and DBPedia (Auer et al., 2007), but even though they are impressively large, their coverage is far from complete. This has motivated research in automatically deriving new facts to extend a manually built knowledge base, by using information from the existing knowledge base, textual mentions of entities, and semi-structured data such as tables and web forms (Nickel et al., 2015).

In this paper we build upon the work of Riedel et al. (2013), which jointly learns continuous representations for knowledge base and textual relations. This common representation in the same vector space can serve as a kind of "universal schema" which admits joint inferences among

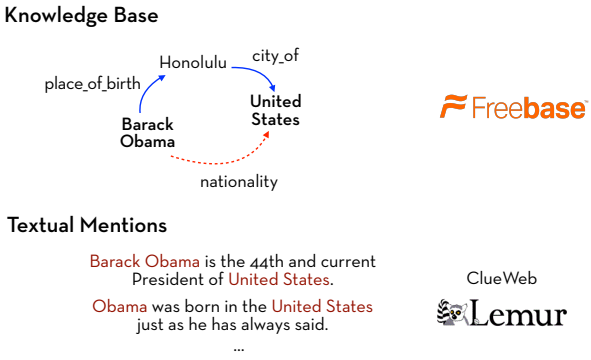

Figure 1: A knowledge base fragment coupled with textual mentions of pairs of entities.

KBs and text. The textual relations represent the relationships between entities expressed in individual sentences (see Figure 1 for an example). Riedel et al. (2013) represented each textual mention of an entity pair by the lexicalized dependency path between the two entities (see Figure 2). Each such path is treated as a separate relation in a combined knowledge graph including both KB and textual relations. Following prior work in latent feature models for knowledge base completion, every textual relation receives its own continuous representation, learned from the pattern of its co-occurrences in the knowledge graph.

However, largely synonymous textual relations often share common sub-structure, and are composed of similar words and dependency arcs. For example, Table 1 shows a collection of dependency paths co-occurring with the *person/organizations_founded* relation.

In this paper we model this sub-structure and share parameters among related dependency paths, using a unified loss function learning entity and relation representations to maximize performance on the knowledge base link prediction task.

We evaluate our approach on the FB15k-237 dataset, a knowledge base derived from the Free-

*This research was conducted during the author's internship at Microsoft Research.

base subset FB15k (Bordes et al., 2013) and filtered to remove highly redundant relations (Toutanova and Chen, 2015). The knowledge base is paired with textual mentions for all entity pairs derived from ClueWeb12[1] with Freebase entity mention annotations (Gabrilovich et al., 2013).

We show that using a convolutional neural network to derive continuous representations for textual relations boosts the overall performance on link prediction, with larger improvement on entity pairs that have textual mentions.

2 Related Work

There has been a growing body of work on learning to predict relations between entities without requiring sentence-level annotations of textual mentions at training time. We group such related work into three groups based on whether KB, text, or both sources of information are used. Additionally, we discuss related work in the area of supervised relation extraction using continuous representations of text, even though we do not use supervision at the level of textual mentions.

Knowledge base completion

Nickel et al. (2015) provide a broad overview of machine learning models for knowledge graphs, including models based on observed graph features such as the path ranking algorithm (Lao et al., 2011), models based on continuous representations (latent features), and model combinations (Dong et al., 2014). These models predict new facts in a given knowledge base, based on information from existing entities and relations. From this line of work, most relevant to our study is prior work evaluating continuous representation models on the FB15k dataset. Yang et al. (2015) showed that a simple variant of a bilinear model DISTMULT outperformed TRANSE (Bordes et al., 2013) and more richly parameterized models on this dataset. We therefore build upon the best performing prior model DISTMULT from this line of work, as well as additional models E and F developed in the context of text-augmented knowledge graphs (Riedel et al., 2013), and extend them to incorporate compositional representations of textual relations.

[1] http://lemurproject.org/clueweb12/FACC1/

Relation extraction using distant supervision

A number of works have focused on extracting new instances of relations using information from textual mentions, without sophisticated modeling of prior knowledge from the knowledge base. Mintz et al. (2009) demonstrated that both surface context and dependency path context were helpful for the task, but did not model the compositional sub-structure of this context. Other work proposed more sophisticated models that reason about sentence-level hidden variables (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012) or model the noise arising from the incompleteness of knowledge bases and text collections (Ritter et al., 2013), *inter alia*. Our work focuses on representing the compositional structure of sentential context for learning joint continuous representations of text and knowledge bases.

Combining knowledge base and text information

A combination of knowledge base and textual information was first shown to outperform either source alone in the framework of path-ranking algorithms in a combined knowledge base and text graph (Lao et al., 2012). To alleviate the sparsity of textual relations arising in such a combined graph, (Gardner et al., 2013; Gardner et al., 2014) showed how to incorporate clusters or continuous representations of textual relations. Note that these vector representations are based on the co-occurrence patterns for the textual relations and not on their compositional structure. Co-occurrence based textual relation representations were also learned in (Neelakantan et al., 2015). Wang et al. (2014a) combined knowledge base and text information by embedding knowledge base entities and the words in their names in the same vector space, but did not model the textual co-occurrences of entity pairs and the expressed textual relations. Weston et al. (2013) combined continuous representations from a knowledge base and textual mentions for prediction of new relations. The two representations were trained independently of each other and using different loss functions, and were only combined at inference time. Additionally, the employed representations of text were non-compositional.

In this work we train continuous representations of knowledge base and textual relations jointly, which allows for deeper interactions between the

sources of information. We directly build on the universal schema approach of Riedel et al. (2013) as well as the universal schema extension of the DISTMULT model mentioned previously, to improve the representations of textual relations by capturing their compositional structure. Additionally, we evaluate the approach on a dataset that contains rich prior information from the training knowledge base, as well as a wealth of textual information from a large document collection.

Continuous representations for supervised relation extraction

In contrast to the work reviewed so far, work on sentence-level relation extraction using direct supervision has focused heavily on representing sentence context. Models using hand-crafted features have evolved for more than a decade, and recently, models using continuous representations have been found to achieve new state-of-the-art performance (Zeng et al., 2014; Gormley et al., 2015). Compared to work on representation learning for sentence-level context, such as this recent work using LSTM models on constituency or dependency trees (Tai et al., 2015), our approach using a one-hidden-layer convolutional neural network is relatively simple. However, even such a simple approach has been shown to be very competitive (Kim, 2014).

3 Models for knowledge base completion

We begin by introducing notation to define the task, largely following the terminology in Nickel et al. (2015). We assume knowledge bases are represented using RDF triples, in the form (*subject*, *predicate*, *object*), where the subject and object are entities and the predicate is the type of relation. For example, the KB fragment shown in Figure 1 is shown as a knowledge graph, where the entities are the nodes, and the relations are shown as directed labeled edges: we see three entities participating in three relation instances indicated by the edges. For brevity, we will denote triples by (e_s, r, e_o), where e_s and e_o denote the subject and object entities, respectively.

The task is, given a training KB consisting of entities with some relations between them, to predict new relations (links) that do not appear in the training KB. More specifically, we will build models that rank candidate entities for given queries $(e_s, r, ?)$ or $(?, r, e_o)$, which ask about the object or subject of a given relation.

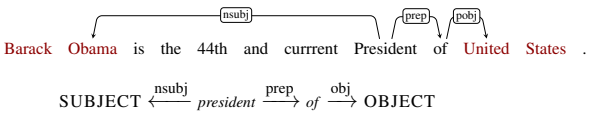

Figure 2: Textual relation extracted from an entity pair mention.

This task setting has been used in models for KB completion previously, e.g. (Dong et al., 2014; Gardner et al., 2014), even though it has not been standard in evaluations of distant supervision for relation extraction (Mintz et al., 2009; Riedel et al., 2013). The advantage of this evaluation setting is that it enables automatic evaluation without requiring humans to label candidate extractions, while making only a *local* closed world assumption for the completeness of the knowledge base — i.e., if one object e_o for a certain subject / relation pair (e_s, r) is present in the knowledge base, it is assumed likely that all other objects (e_s, r, e'_o) will be present. Such an assumption is particularly justified for nearly functional relations.

To incorporate textual information, we follow prior work (Lao et al., 2012; Riedel et al., 2013) and represent both textual and knowledge base relations in a single graph of "universal" relations. The textual relations are represented as full lexicalized dependency paths, as illustrated in Figure 2. An instance of the textual relation SUBJECT $\xleftarrow{\text{nsubj}}$ *president* $\xrightarrow{\text{prep}}$ *of* $\xrightarrow{\text{obj}}$ OBJECT connecting the entities BARACK OBAMA and UNITED STATES, is added to the knowledge graph based on this sentential occurrence.

To present the models for knowledge base completion based on such combined knowledge graphs, we first introduce some notation. Let $\mathcal{E}$ denote the set of entities in the knowledge graph and let $\mathcal{R}$ denote the set of relation types. We denote each possible triple as $T = (e_s, r, e_o)$ where $e_s, e_o \in \mathcal{E}, r \in \mathcal{R}$, and model its presence with a binary random variable $y_T \in \{0, 1\}$ which indicates whether the triple exists. The models we build score possible triples (e_s, r, e_o) using *continuous representations* (*latent features*) of the three elements of the triple. The models use scoring function $f(e_s, r, e_o)$ to represent the model's confidence in the existence of the triple. We present the models and then the loss function used to train

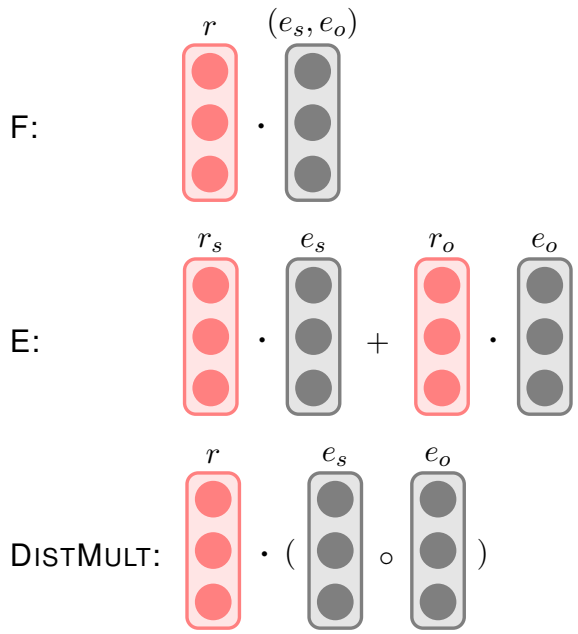

Figure 3: The continuous representations for model F, E and DISTMULT.

their parameters.

3.1 Basic Models

We begin with presenting the three models from prior work that this research builds upon. They all learn latent continuous representations of relations and entities or entity pairs, and score possible triples based on the learned continuous representations. Each of the models can be defined on a knowledge graph containing entities and KB relations only, or on a knowledge graph additionally containing textual relations. We use models F and E from (Riedel et al., 2013) where they were used for a combined KB+text graph, and model DISTMULT from (Yang et al., 2015), which was originally used for a knowledge graph containing only KB relations.

As shown in Figure 3, model F learns a K-dimensional latent feature vector for each candidate entity pair (e_s, e_o), as well as a same-dimensional vector for each relation r, and the scoring function is simply defined as their inner product: $f(e_s, r, e_o) = v(r)^\intercal v(e_s, e_o)$. Therefore, different pairs sharing the same entity would not share parameters in this model.

Model E does not have parameters for entity pairs, and instead has parameters for individual entities. It aims to capture the compatibility between entities and the subject and object positions of relations. For each relation type r, the model learns two latent feature vectors $v(r_s)$ and $v(r_o)$ of dimension K. For each entity (node) e_i, the model also learns a latent feature vector of the same dimensionality. The score of a candidate triple (e_s, r, e_o) is defined as $f(e_s, r, e_o) = v(r_s)^\intercal v(e_s) + v(r_o)^\intercal v(e_o)$. It can be seen that when a subject entity is fixed in a query $(e_s, r, ?)$, the ranking of candidate object entity fillers according to f does not depend on the subject entity but only on the relation type r.

The third model DISTMULT, is a special form of a bilinear model like RESCAL (Nickel et al., 2011), where the non-diagonal entries in the relation matrices are assumed to be zero. This model was proposed in Yang et al. (2015) and was shown to outperform prior work on the FB15k dataset. In this model, each entity e_i and each relation r is assigned a latent feature vector of dimension K. The score of a candidate triple (e_s, r, e_o) is defined as $f(e_s, r, e_o) = v(r)^\intercal \left(v(e_s) \circ v(e_o)\right)$, where $\circ$ denotes the element-wise vector product. In this model, entity pairs which share an entity also share parameters, and the ranking of candidate objects for queries $(e_s, r, ?)$ depends on the subject entity.

Denote $N_e = |\mathcal{E}|, N_r = |\mathcal{R}|$, and K = dimension of latent feature vectors, then model E has $KN_e + 2KN_r$ parameters and model DISTMULT has $KN_e + KN_r$ parameters. Model F has $KN_e^2 + KN_r$ parameters, although most entity pairs will not co-occur in the knowledge base or text.

In the basic models, knowledge base and textual relations are treated uniformly, and each textual relation receives its own latent representation of dimensionality K. When textual relations are added to the training knowledge graph, the total number of relations $|\mathcal{R}|$ grows substantially (it increases from 237 to more than 2.7 million for the dataset in this study), resulting in a substantial increase in the total number of independent parameters.

Note that in all of these models queries about the arguments of knowledge base relations $(e_s, r, ?)$ are answered by scoring functions looking only at the entity and KB relation representations, without using representations of textual mentions. The textual mention information and representations are only used at training time to improve the learned representations of KB relations and entities.

3.2 CONV: Compositional Representations of Textual Relations

In the standard latent feature models discussed above, each textual relation is treated as an atomic unit receiving its own set of latent features. However, many textual relations differ only slightly in the words or dependency arcs used to express the relation. For example, Table 1 shows several textual patterns that co-occurr with the relation *person/organizations_founded* in the training KB. While some dependency paths occur frequently, many very closely related ones have been observed only once. The statistical strength of the model could be improved if similar dependency paths have a shared parameterization. We build on work using similar intuitions for other tasks and learn compositional representations of textual relations based on their internal structure, so that the derived representations are accurate for the task of predicting knowledge base relations.

We use a convolutional neural network applied to the lexicalized dependency paths treated as a sequence of words and dependency arcs with direction. Figure 4 depicts the neural network architecture. In the first layer, each word or directed labeled arc is mapped to a continuous representation using an embedding matrix $\mathbf{V}$. In the hidden layer, every window of three elements is mapped to a hidden vector using position-specific maps $\mathbf{W}$, a bias vector $\mathbf{b}$, and a $\tanh$ activation function. A max-pooling operation over the sequence is applied to derive the final continuous representation for the dependency path.

The CONV representation of textual relations can be used to augment any of the three basic models. The difference between a basic model and its CONV-augmented variant is in the parameterization of textual mentions. The basic models learn distinct latent feature vectors of dimensionality K for all textual relation types, whereas the CONV models derive the K-dimensional latent feature vectors for textual relation types as the activation at the top layer of the convolutional network in Figure 4, given the corresponding lexicalized dependency path as input.

3.3 Training loss function

All basic and CONV-augmented models use the same training loss function. Our loss function is motivated by the link prediction task and the performance measures used. As previously mentioned, the task is to predict the subject or object entity for given held-out triples (e_s, r, e_o), i.e., to rank all entities with respect to their likelihood of filling the respective position in the triple[2]. We would thus like the model to score correct triples (e_s, r, e_o) higher than incorrect triples (e', r, e_o) and (e_s, r, e') which differ from the correct triple by one entity. Several approaches (Nickel et al., 2015) use a margin-based loss function. We use an approximation to the negative log-likelihood of the correct entity filler instead[3]. We define the conditional probabilities $p(e_o|e_s, r)$ and $p(e_s|r, e_o)$ for object and subject entities given the relation and the other argument as follows:

$$p(e_o|e_s, r; \Theta) = \frac{e^{f(e_s,r,e_o;\Theta)}}{\sum_{e' \in Neg(e_s,r,?)} e^{f(e_s,r,e';\Theta)}}$$

Conditional probabilities for subject entities $p(e_s|e_o, r; \Theta)$ are defined analogously. Here Θ denotes all the parameters of latent features. The denominator is defined using a set of entities that do not fill the object position in any relation triple $(e_s, r, ?)$ in the training knowledge graph. Since the number of such entities is impractically large, we sample negative triples from the full set. We also limit the candidate entities to ones that have types consistent with the position in the relation triple (Chang et al., 2014; Yang et al., 2015), where the types are approximated following Toutanova and Chen (2015). Additionally, since the task of predicting textual relations is auxiliary to the main task, we use a weighting factor τ for the loss on predicting the arguments of textual relations (Toutanova and Chen, 2015).

Denote $\mathcal{T}$ as a set of triples, we define the loss $L(\mathcal{T}; \Theta)$ as:

$$\begin{aligned} L(\mathcal{T}; \Theta) \quad = \quad & - \sum_{(e_s,r,e_o) \in \mathcal{T}} \log p(e_o|e_s, r; \Theta) \\ & - \sum_{(e_s,r,e_o) \in \mathcal{T}} \log p(e_s|e_o, r; \Theta) \end{aligned}$$

Let $\mathcal{T}_{\text{KB}}$ and $\mathcal{T}_{\text{text}}$ represent the set of knowledge base triples and textual relation triples respectively. The final training loss function is de-

[2] Our experimental comparison focuses on predicting object entities only, but we consider both argument types in the training loss function.

[3] Note that both margin-based and likelihood-based loss functions are susceptible to noise from potential selection of false negative examples. An empirical comparison of training loss functions would be interesting.

Textual Pattern	Count
SUBJECT $\xrightarrow{\text{appos}}$ founder $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	12
SUBJECT $\xleftarrow{\text{nsubj}}$ co-founded $\xrightarrow{\text{dobj}}$ OBJECT	3
SUBJECT $\xrightarrow{\text{appos}}$ co-founder $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	3
SUBJECT $\xrightarrow{\text{conj}}$ co-founder $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	3
SUBJECT $\xleftarrow{\text{pobj}}$ with $\xleftarrow{\text{prep}}$ co-founded $\xrightarrow{\text{dobj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{nsubj}}$ signed $\xrightarrow{\text{xcomp}}$ establishing $\xrightarrow{\text{dobj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{pobj}}$ with $\xleftarrow{\text{prep}}$ founders $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	2
SUBJECT $\xrightarrow{\text{appos}}$ founders $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{nsubj}}$ one $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ founders $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{nsubj}}$ founded $\xrightarrow{\text{dobj}}$ production $\xrightarrow{\text{conj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{appos}}$ partner $\xleftarrow{\text{pobj}}$ with $\xleftarrow{\text{prep}}$ founded $\xrightarrow{\text{dobj}}$ production $\xrightarrow{\text{conj}}$ OBJECT	2
SUBJECT $\xleftarrow{\text{pobj}}$ by $\xleftarrow{\text{prep}}$ co-founded $\xleftarrow{\text{rcmod}}$ OBJECT	1
SUBJECT $\xleftarrow{\text{nn}}$ co-founder $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	1
SUBJECT $\xrightarrow{\text{dep}}$ co-founder $\xrightarrow{\text{prep}}$ of $\xrightarrow{\text{pobj}}$ OBJECT	1
SUBJECT $\xleftarrow{\text{nsubj}}$ helped $\xrightarrow{\text{xcomp}}$ establish $\xrightarrow{\text{dobj}}$ OBJECT	1
SUBJECT $\xleftarrow{\text{nsubj}}$ signed $\xrightarrow{\text{xcomp}}$ creating $\xrightarrow{\text{dobj}}$ OBJECT	1

Table 1: Textual patterns occurring with entity pairs in a *person/organizations_founded* relationship. The count indicates the number of training set instances that have this KB relation, which co-occur with each textual pattern.

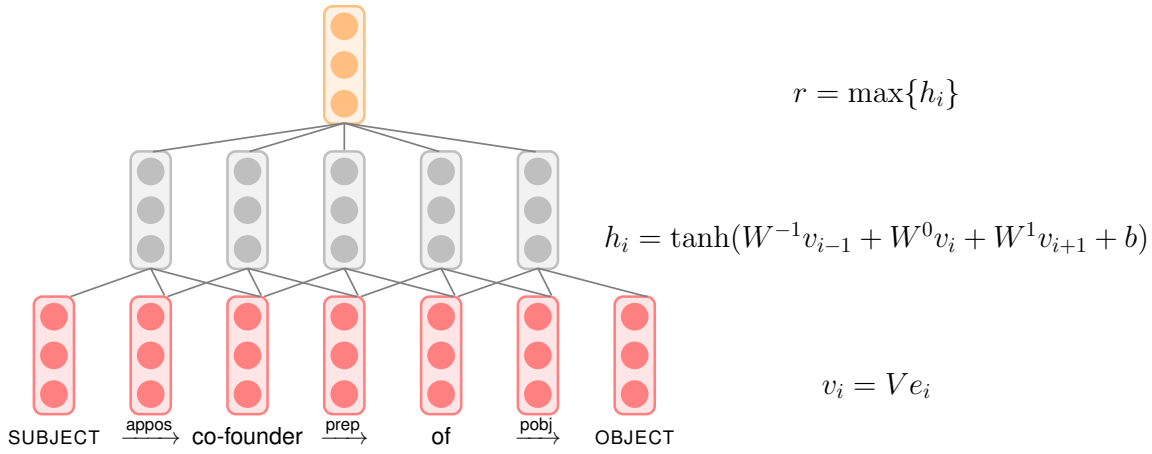

Figure 4: The convolutional neural network architecture for representing textual relations.

fined as:

$$L(\mathcal{T}_{\text{KB}}; \Theta) + \tau L(\mathcal{T}_{\text{text}}; \Theta) + \lambda \|\Theta\|^2,$$

where λ is the regularization parameter, and τ is the weighing factor of the textual relations.

The parameters of all models are trained using a batch training algorithm. The gradients of the basic models are straightforward to compute, and the gradients of the convolutional network parameters for the CONV-augmented models are also not hard to derive using back-propagation.

4 Experiments

Dataset and Evaluation Protocol

We use the FB15k-237 [4] dataset, which is a subset of FB15k (Bordes et al., 2013) that excludes redundant relations and direct training links for held-out triples, with the goal of making the task more realistic (Toutanova and Chen, 2015). The FB15k dataset has been used in multiple studies on knowledge base completion (Wang et al., 2014b; Yang et al., 2015). Textual relations for

[4] Check the first author's website for a release of the dataset.

FB15k-237 are extracted from 200 million sentences in the ClueWeb12 corpus coupled with Freebase mention annotations (Gabrilovich et al., 2013), and include textual links of all co-occurring entities from the KB set. After pruning[5], there are 2.7 million unique textual relations that are added to the knowledge graph. The set of textual relations is larger than the set used in Toutanova and Chen (2015) (25,000 versus 2.7 million), leading to improved performance.

The number of relations and triples in the training, validation and test portions of the data are given in Table 2. The two rows list statistics for the KB and text portions of the data separately. The 2.7 million textual relations occur in 3.9 million text triples. Almost all entities occur in textual relations (13,937 out of 14,541). The numbers of triples for textual relations are shown as zero for the validation and test sets because we don't evaluate on prediction of textual relations (all text triples are used in training). The percentage of KB triples that have textual relations for their pair of entities is 40.5% for the training, 26.6% for the validation, and 28.1% for the test set. While 26.6% of the validation set triples have textual mentions, the percentage with textual relations that have been seen in the training set is 18.4%. Having a mention increases the chance that a random entity pair has a relation from 0.1% to 5.0% — a fifty-fold increase.

Given a set of triples in a set disjoint from a training knowledge graph, we test models on predicting the object of each triple, given the subject and relation type. We rank all entities in the training knowledge base in order of their likelihood of filling the argument position. We report the mean reciprocal rank (MRR) of the correct entity, as well as HITS@10 — the percentage of test triples for which the correct entity is ranked in the top 10. We use *filtered* measures following the protocol proposed in Bordes et al. (2013) — that is, when we rank entities for a given position, we remove all other entities that are known to be part of an existing triple in the training, validation, or test set. This avoids penalizing the model for ranking other correct fillers higher than the tested entity.

[5]The full set of 37 million textual patterns connecting the entity pairs of interest was pruned based on the count of patterns and their tri-grams, and their precision in indicating that entity pairs have KB relations.

Implementation details

We used a value of $\lambda = 1$ for the weight of the L_2 penalty for the main results in Table 3, and present some results on the impact of λ at the end of this section. We used batch optimization after initial experiments with AdaGrad showed inferior performance. L-BFGS (Liu and Nocedal, 1989) and RProp (Riedmiller and Braun, 1993) were found to converge to similar function values, with RProp converging significantly faster. We thus used RProp for optimization. We initialized the KB+text models from the KB-only models and also from random initial values (sampled from a Gaussian distribution), and stopped optimization when the overall MRR on the validation set decreased. For each model type, we chose the better of random and KB-only initialization. The word embeddings in the CONV models were initialized using the 50-dimensional vectors from Turian et al. (2010) in the main experiments, with a slight positive impact. The effect of initialization is discussed at the end of the section.

The number of negative examples for each triple was set to 200. Performance improved substantially when the number of negative examples was increased and reached a plateau around 200. We chose the optimal number of latent feature dimensions via a grid search to optimize MRR on the validation set, testing the values 5, 10, 15, 35, 50, 100, 200 and 500. We also performed a grid search over the values of the parameter τ, testing values in the set $\{0.01, 0.1, 0.25, 0.5, 1\}$. The best dimension for latent feature vectors was 10 for most KB-only models (not including model F), and 5 for the two model configurations including F. We used $K = 10$ for all KB+text models, as higher dimension was also not helpful for them.

Experimental results

In Table 3 we show the performance of different models and their combinations[6], both when using textual mentions (*KB+text*), and when using only knowledge base relations (*KB only*). In the KB+text setting, we evaluate the contribution of the CONV representations of the textual relations. The upper portion of the Table shows the performance of models that have been trained using knowledge graphs including only knowledge

[6]Different models are combined by simply defining a combined scoring function which adds the scores from individual models. Combined models are trained jointly.

	# Relations	# Entities	# Triples in Train / Validation / Test
KB	237	14,541	272,115 / 17,535 / 20, 466
Text	2,740k	13,937	3,978k / 0 / 0

Table 2: The statistics of dataset FB15k-237.

Model	Overall		With mentions		Without mentions	
	MRR	HITS@10	MRR	HITS@10	MRR	HITS@10
KB only						
F	16.9	24.5	26.4	49.1	13.3	15.5
E	33.2	47.6	25.5	37.8	36.0	51.2
DISTMULT	35.7	52.3	26.0	39.0	39.3	57.2
E+DISTMULT	**37.3**	**55.2**	**28.6**	**42.9**	40.5	**59.8**
F+E+DISTMULT	33.8	50.1	15.0	26.1	**40.7**	59.0
KB and text						
F ($\tau = 1$)	19.4	27.9	**35.4**	61.6	13.4	15.5
CONV-F ($\tau = 1$)	19.2	28.4	34.9	**63.7**	13.3	15.4
E ($\tau = 0$)	33.2	47.6	25.5	37.8	36.0	51.2
CONV-E ($\tau = 0$)	33.2	47.6	25.5	37.8	36.0	51.2
DISTMULT ($\tau = 0.01$)	36.1	52.7	26.5	39.5	39.6	57.5
CONV-DISTMULT ($\tau = 0.25$)	36.6	53.5	28.3	43.4	39.7	57.2
E + DISTMULT ($\tau = 0.01$)	37.7	55.7	28.9	43.4	40.9	60.2
CONV-E + CONV-DISTMULT ($\tau = 0.25$)	**40.1**	**58.1**	33.9	49.9	**42.4**	**61.1**

Table 3: Results on FB15k-237 for KB only and KB+text inference, with basic models versus the proposed CONV-augmented models. The values of the hyper-parameter τ (as shown in the Table) were chosen to maximize MRR on the validation set. The reported numbers were obtained for the test set.

base relations, and are not using any information from textual mentions. The lower portion of the Table shows the performance when textual relations are added to the training knowledge graph and the corresponding training loss function. Note that all models predict based on the learned knowledge base relation and entity representations, and the textual relations are only used at training time when they can impact these representations.

The performance of all models is shown as an overall MRR (scaled by 100) and HITS@10, as well as performance on the subset of triples that have textual mentions (column *With mentions*), and ones that do not (column *Without mentions*). Around 28% of the test triples have mentions and contribute toward the measures in the *With mentions* column, and the other 72% of the test triples contribute to the *Without mentions* column.

For the KB-only models, we see the performance of each individual model F, E, and DISTMULT. Model F was the best performing single model from (Riedel et al., 2013), but it does not perform well when textual mentions are not used. In our implementation of model F, we created entity pair parameters only for entity pairs that co-occur in the text data (Riedel et al. (2013) also trained pairwise vectors for co-occuring entities only, but all of the training and test tuples in their study were co-occurring)[7]. Without textual information, model F is performing essentially randomly, because entity pairs in the test sets do not occur in training set relations (by construction of the dataset). Model E is able to do surprisingly well, given that it is making predictions for each object position of a relation without considering the given subject of the relation. DISTMULT is the best performing single model. Unlike model F, it is able to share parameters among entity pairs with common subject or object entities, and, unlike model E, it captures some dependencies between the subject and object entities of a relation. The combination of models E+DISTMULT improves performance, but combining model F with the other two is not helpful.

The lower portion of Table 3 shows results when textual relations are added to the training knowledge graph. The basic models treat the textual relations as atomic and learn a separate latent feature vector for each textual relation. The CONV- models use the compositional representations of tex-

[7]Learning entity pair parameters for all entity pairs would result in 2.2 billion parameters for vectors with dimensionality 10 for our dataset. This was infeasible and was also not found useful based on experiments with vectors of lower dimensionality.

tual relations learned using the convolutional neural network architecture shown in Figure 4. We show the performance of each individual model and its corresponding variant with a CONV parameterization. For each model, we also show the optimal value of τ, the weight of the textual relations loss. Model F is able to benefit from textual relations and its performance increases by 2.5 points in MRR, with the gain in performance being particularly large on test triples with textual mentions. Model F is essentially limiting its space of considered argument fillers to ones that have co-occurred with the given subject entity. This gives it an advantage on test triples with textual mentions, but model F still does relatively very poorly overall when taking into account the much more numerous test triples without textual mentions. The CONV parameterization performs slightly worse in MRR, but slightly better in HITS@10, compared to the atomic parameterization. For model E and its CONV variant, we see that text does not help as its performance using text is the same as that when not using text and the optimal weight of the text is zero. Model DISTMULT benefits from text, and its convolutional text variant CONV-DISTMULT outperforms the basic model, with the gain being larger on test triples with mentions.

The best model overall, as in the KB-only case, is E+DISTMULT. The basic model benefits from text slightly and the model with compositional representations of textual patterns CONV-E+CONV-DISTMULT, improves the performance further, by 2.4 MRR overall, and by 5 MRR on triples with textual mentions. It is interesting that the text and the compositional representations helped most for this combined model. One hypothesis is that model E, which provides a prior over relation arguments, is needed in combination with DISTMULT to prevent the prediction of unlikely arguments based on noisy inference from textual patterns and their individual words and dependency links.

Hyperparameter Sensitivity

To gain insight into the sensitivity of the model to hyper-parameters and initialization, we report on experiments starting with the best model CONV-E + CONV-DISTMULT from Table 3 and varying one parameter at a time. This model has weight of the textual relations loss $\tau = 0.25$, weight of the L_2 penalty $\lambda = 1$, convolution window size of three, and is initialized randomly for the entity and KB relation vectors, and from pre-trained embeddings for word vectors (Turian et al., 2010). The overall MRR of the model is 40.4 on the validation set (test results are shown in the Table).

When the weight of τ is changed to 1 (i.e., equal contribution of textual and KB relations), the overall MRR goes down to 39.6 from 40.4, indicating the usefulness of weighting the two kinds of relations non-uniformly. When λ is reduced to 0.04, MRR is 40.0 and when λ is increased to 25, MRR goes down to 38.9. This indicates the L_2 penalty hyper-parameter has a large impact on performance. When we initialize the word embeddings randomly instead of using pre-trained word vectors, performance drops only slightly to 40.3. If we initialize from a model trained using KB-only information, performance goes down substantially to 38.7. This indicates that initialization is important and there is a small gain from using pre-trained word embeddings. There was a drop in performance to MRR 40.2 when using a window size of one for the convolutional architecture in Figure 4, and an increase to 40.6 when using a window size of five.

5 Conclusion and Future Work

Here we explored an alternative representation of textual relations for latent feature models that learn to represent knowledge base and textual relations in the same vector space. We showed that given the large degree of sharing of sub-structure in the textual relations, it was beneficial to compose their continuous representations out of the representations of their component words and dependency arc links. We applied a convolutional neural network model and trained it jointly with a model mapping entities and knowledge base relations to the same vector space, obtaining substantial improvements over an approach that treats the textual relations as atomic units having independent parameterization.

Acknowledgements

We would like to thank the anonymous reviewers for their suggestions, and Jianfeng Gao, Scott Wen-tau Yih, and Wei Xu for useful discussions.

References

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

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

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 (NIPS)*.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora, Version 1 (release date 2013-06-26, format version 1, correction level 0).

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Association for Computational Linguistics (ACL)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *ACL*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Martin Riedmiller and Heinrich Braun. 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE.

Alan Ritter, Luke Zettlemoyer, Oren Etzioni, et al. 2013. Modeling missing data in distant supervision for information extraction. *Transactions of the Association for Computational Linguistics*, 1:367–378.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*.

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*.

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014a. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics*, pages 1591–1601.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*.

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