

Quantifying Similarity between Relations with Fact Distribution

Weize Chen¹ Hao Zhu^{1,2} Xu Han¹ Zhiyuan Liu¹ Maosong Sun¹

¹ Department of Computer Science and Technology, Tsinghua University, Beijing, China

State Key Lab on Intelligent Technology and Systems

Institute for Artificial Intelligence

{wei10, hanxu17}@mails.tsinghua.edu.cn {liuzy, sms}@tsinghua.edu.cn

² Carnegie Mellon University, Pittsburgh, PA, USA

zhuhao@cmu.edu

Abstract

We introduce a conceptually simple and effective method to quantify the *similarity* between relations in knowledge bases. Specifically, our approach is based on the divergence between the conditional probability distributions over entity pairs. In this paper, these distributions are parameterized by a very simple neural network. Although computing the exact similarity is intractable, we provide a sampling-based method to get a good approximation.

We empirically show the outputs of our approach significantly correlate with human judgments. By applying our method to various tasks, we also find that (1) our approach could effectively detect redundant relations extracted by open information extraction (Open IE) models, that (2) even the most competitive models for relational classification still make mistakes among very similar relations, and that (3) our approach could be incorporated into negative sampling and softmax classification to alleviate these mistakes. The source code and experiment details of this paper can be obtained from <https://github.com/thunlp/relation-similarity>.

1 Introduction

Relations¹, representing various types of connections between entities or arguments, are the core of expressing relational facts in most general knowledge bases (KBs) (Suchanek et al., 2007; Bollacker et al., 2008). Hence, identifying relations is a crucial problem for several information extraction tasks. Although considerable effort has been devoted to these tasks, some nuances between similar relations

Sentence	The crisis didn't influence his two daughters OBJ and SUBJ.
Correct	per:siblings
Predicted	per:parents
Similarity Rank	2

Table 1: An illustration of the errors made by relation extraction models. The sentence contains obvious patterns indicating the two persons are siblings, but the model predicts it as parents. We introduce an approach to measure the similarity between relations. Our result shows “siblings” is the second most similar one to “parents”. By applying this approach, we could analyze the errors made by models, and help reduce errors.

are still overlooked, (Table 1 shows an example); on the other hand, some distinct surface forms carrying the same relational semantics are mistaken as different relations. These severe problems motivate us to quantify the similarity between relations in a more effective and robust method.

In this paper, we introduce an adaptive and general framework for measuring similarity of the pairs of relations. Suppose for each relation r , we have obtained a conditional distribution, $P(h, t \mid r)$ ($h, t \in \mathcal{E}$ are head and tail entities, and $r \in \mathcal{R}$ is a relation), over all head-tail entity pairs given r . We could quantify similarity between a pair of relations by the divergence between the conditional probability distributions given these relations. In this paper, this conditional probability is given by a simple feed-forward neural network, which can capture the dependencies between entities conditioned on specific relations. Despite its simplicity, the proposed network is expected to cover various facts, even if the facts are not used for training, owing to the good generalizability of neural networks. For example, our network will assign a fact a higher probability if it is “logical”: e.g., the network might prefer an athlete has the same nationality as same as his/her national team rather than other nations.

Author contributions: Hao Zhu designed the research; Weize Chen prepared the data, and organized data annotation; Hao Zhu and Xu Han designed the experiments; Weize Chen performed the experiments; Hao Zhu, Weize Chen and Xu Han wrote the paper; Zhiyuan Liu and Maosong Sun proofread the paper. Zhiyuan Liu is the corresponding author.

¹Sometimes relations are also named properties.

Intuitively, two similar relations should have similar conditional distributions over head-tail entity pairs $P(h, t | r)$, e.g., the entity pairs associated with *be trade to* and *play for* are most likely to be athletes and their clubs, whereas those associated with *live in* are often people and locations. In this paper, we evaluate the similarity between relations based on their conditional distributions over entity pairs. Specifically, we adopt Kullback–Leibler (KL) divergence of both directions as the metric. However, computing exact KL requires iterating over the whole entity pair space $\mathcal{E} \times \mathcal{E}$, which is quite intractable. Therefore, we further provide a sampling-based method to approximate the similarity score over the entity pair space for computational efficiency.

Besides developing a framework for assessing the similarity between relations, our second contribution is that we have done a survey of applications. We present experiments and analysis aimed at answering five questions:

(1) How well does the computed similarity score correlate with human judgment about the similarity between relations? How does our approach compare to other possible approaches based on other kinds of relation embeddings to define a similarity? (§3.4 and §5)

(2) Open IE models inevitably extract many redundant relations. How can our approach help reduce such redundancy? (§6)

(3) To which extent, quantitatively, does best relational classification models make errors among similar relations? (§7)

(4) Could similarity be used in a heuristic method to enhance negative sampling for relation prediction? (§8)

(5) Could similarity be used as an adaptive margin in softmax-margin training method for relation extraction? (§9)

Finally, we conclude with a discussion of valid extensions to our method and other possible applications.

2 Learning Head-Tail Distribution

Just as introduced in §1, we quantify the similarity between relations by their corresponding head-tail entity pair distributions. Consider the typical case that we have got numbers of facts, but they are still sparse among all facts in the real world. How could we obtain a well-generalized distribution over the whole space of possible triples beyond the

training facts? This section proposes a method to parameterize such a distribution.

2.1 Formal Definition of Fact Distribution

A *fact* is a triple $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where h and t are called *head* and *tail* entities, r is the relation connecting them, \mathcal{E} and \mathcal{R} are the sets of entities and relations respectively. We consider a score function $F_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$ maps all triples to a scalar value. As a special case, the function can be factorized into the sum of two parts: $F_\theta(h, t; r) \triangleq u_{\theta_1}(h; r) + u_{\theta_2}(t; h, r)$. We use F_θ to define the unnormalized probability.

$$\tilde{P}_\theta(h, t | r) \triangleq \exp F_\theta(h, r; t) \quad (1)$$

for every triple (h, r, t) . The real parameter θ can be adjusted to obtain difference distributions over facts.

In this paper, we only consider *locally normalized* version of F_θ :

$$\begin{aligned} u_{\theta_1}(h; r) &= \log \frac{\exp \tilde{u}_{\theta_1}(h; r)}{\sum_{h'} \exp \tilde{u}_{\theta_1}(h'; r)}, \\ u_{\theta_2}(t; h, r) &= \log \frac{\exp \tilde{u}_{\theta_2}(t; h, r)}{\sum_{t'} \exp \tilde{u}_{\theta_2}(t'; h, r)}, \end{aligned} \quad (2)$$

where \tilde{u}_{θ_1} and \tilde{u}_{θ_2} are directly parameterized by feed-forward neural networks. Through local normalization, $\tilde{P}_\theta(h, t | r)$ is naturally a valid probability distribution, as the partition function $\sum_{h, t} \exp F_\theta(h, t; r) = 1$. Therefore, $P_\theta(h, t | r) = \tilde{P}_\theta(h, t | r)$.

2.2 Neural architecture design

Here we introduce our special design of neural networks. For the first part and the second part, we implement the scoring functions introduced in equation (2) as

$$\begin{aligned} \tilde{u}_{\theta_1}(h; r) &= \text{MLP}_{\theta_1}(\mathbf{r})^\top \mathbf{h}, \\ \tilde{u}_{\theta_2}(t; h, r) &= \text{MLP}_{\theta_2}([\mathbf{h}; \mathbf{r}])^\top \mathbf{t}, \end{aligned} \quad (3)$$

where each MLP_θ represents a multi-layer perceptron composed of layers like $\mathbf{y} = \text{relu}(\mathbf{W}\mathbf{x} + \mathbf{b})$, $\mathbf{h}, \mathbf{r}, \mathbf{t}$ are embeddings of h, r, t , and θ includes weights and biases in all layers.

2.3 Training

Now we discuss the method to perform training. In this paper, we consider joint training. By minimizing the loss function, we compute the model parameters θ^* :

$$\begin{aligned} \theta^* &= \argmin_{\theta} \mathcal{L}(G) \\ &= \argmin_{\theta} \sum_{(h, r, t) \in G} -\log P_\theta(h, t | r), \end{aligned} \quad (4)$$

where $G \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of triples.² The whole set of parameters, $\theta = \{\theta_1, \theta_2, \{e, \forall e \in \mathcal{E}\}, \{r, \forall r \in \mathcal{R}\}\}$. We train these parameters by Adam optimizer (Kingma and Ba, 2014). Training details are shown in Appendix C.

3 Quantifying Similarity

So far, we have talked about how to use neural networks to approximate the natural distribution of facts. The center topic of our paper, quantifying similarity, will be discussed in detail in this section.

3.1 Relations as Distributions

In this paper, we provide a probability view of relations by representing relation r as a probability distribution $P_{\theta^*}(h, t | r)$. After training the neural network on a given set of triples, the model is expected to generalize well on the whole $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ space.

Note that it is very easy to calculate $P_{\theta^*}(h, t | r)$ in our model thanks to local normalization (equation (2)). Therefore, we can compute it by

$$P_{\theta^*}(h, t | r) = \exp(u_{\theta_1}(h; r) + u_{\theta_2}(t; h, r)). \quad (5)$$

3.2 Defining Similarity

As the basis of our definition, we hypothesize that the similarity between $P_{\theta^*}(h, t | r)$ reflects the similarity between relations.³ For example, if the conditional distributions of two relations put mass on similar entity pairs, the two relations should be quite similar. If they emphasize different ones, the two should have some differences in meaning.

Formally, we define the similarity between two relations as a function of the divergence between the distributions of corresponding head-tail entity pairs:

$$S(r_1, r_2) = g\left(D_{\text{KL}}(P_{\theta^*}(h, t | r_1) || P_{\theta^*}(h, t | r_2)), D_{\text{KL}}(P_{\theta^*}(h, t | r_2) || P_{\theta^*}(h, t | r_1))\right), \quad (6)$$

where $D_{\text{KL}}(\cdot || \cdot)$ denotes Kullback–Leibler divergence,

$$D_{\text{KL}}(P_{\theta^*}(h, t | r_1) || P_{\theta^*}(h, t | r_2)) = \mathbb{E}_{h, t \sim P_{\theta^*}(h, t | r_1)} \log \frac{P_{\theta^*}(h, t | r_1)}{P_{\theta^*}(h, t | r_2)} \quad (7)$$

²In our applications, the set of triples could be a knowledge base or a set of triples in the training set etc.

³§5 provides empirical results to corroborate this hypothesis.

vice versa, and function $g(\cdot, \cdot)$ is a symmetrical function. To keep the coherence between semantic meaning of “similarity” and our definition, g should be a monotonically decreasing function. Through this paper, we choose to use an exponential family⁴ composed with max function, i.e., $g(x, y) = e^{-\max(x, y)}$. Note that by taking both sides of KL divergence into account, our definition incorporates both the entity pairs with high probability in r_1 and r_2 . Intuitively, if $P_{\theta^*}(h, t | r_1)$ mainly distributes on a proportion of entities pairs that $P_{\theta^*}(h, t | r_2)$ emphasizes, r_1 is only hyponymy of r_2 . Considering both sides of KL divergence could help model yield more comprehensive consideration. We will talk about the advantage of this method in detail in §3.4.

3.3 Calculating Similarity

Just as introduced in §1, it is intractable to compute similarity exactly, as involving $\mathcal{O}(|\mathcal{E}|^2)$ computation. Hence, we consider the monte-carlo approximation:

$$\begin{aligned} & D_{\text{KL}}(P_{\theta^*}(h, t | r_1) || P_{\theta^*}(h, t | r_2)) \\ &= \mathbb{E}_{h, t \sim P_{\theta^*}(h, t | r_1)} \log \frac{P_{\theta^*}(h, t | r_1)}{P_{\theta^*}(h, t | r_2)} \\ &\approx \frac{1}{|\mathcal{S}|} \sum_{h, t \in \mathcal{S}} \log \frac{P_{\theta^*}(h, t | r_1)}{P_{\theta^*}(h, t | r_2)}, \end{aligned} \quad (8)$$

where \mathcal{S} is a list of entity pairs sampled from $P_{\theta^*}(h, t | r_1)$. We use sequential sampling⁵ to gain \mathcal{S} , which means we first sample h given r from $u_{\theta_1}(h; r)$, and then sample t given h and r from $u_{\theta_2}(t; h, r)$.⁶

3.4 Relationship with other metrics

Previous work proposed various methods for representing relations as vectors (Bordes et al., 2013; Yang et al., 2015), as matrices (Nickel et al., 2011), even as angles (Sun et al., 2019), etc. Based on each of these representations, one could easily define various similarity quantification methods.⁷ We show in Table 2 the best one of them in each category of relation presentation.

Here we provide two intuitive reasons for using our proposed probability-based similarity: (1)

⁴We view KL divergences as energy functions.

⁵Sampling h and t at the same time requires $\mathcal{O}(|\mathcal{E}|^2)$ computation, while sequential sampling requires only $\mathcal{O}(|\mathcal{E}|)$ computation.

⁶It seems to be a non-symmetrical method, and sampling from the mixture of both forward and backward should yield a better result. Surprisingly, in practice, sampling from single direction works just as well as from both directions.

⁷Taking the widely used vector representations as an example, we can define the similarity between relations based on cosine distance, dot product distance, L1/L2 distance, etc.

Relation Representation	Method	Similarity Quantification
Vectors	TransE (Bordes et al., 2013)	$S(r_1, r_2) = \exp(\mathbf{r}_1^\top \mathbf{r}_2 / \ \mathbf{r}_1\ _2 \ \mathbf{r}_2\ _2)$
Vectors	DistMult (Yang et al., 2015)	$S(r_1, r_2) = \exp(\mathbf{r}_1^\top \mathbf{r}_2 / \ \mathbf{r}_1\ _2 \ \mathbf{r}_2\ _2)$
Matrices	RESICAL (Nickel et al., 2011)	$S(r_1, r_2) = \exp(\ M_{r_1} - M_{r_2}\ _F)$
Angles	RotatE (Sun et al., 2019)	$S(r_1, r_2) = \exp(-\sum_{i=1}^n \mathbf{r}_{1,i} - \mathbf{r}_{2,i} _1)$
Probability Distribution	Ours	equation (6)

Table 2: Methods to define a similarity function with different types of relation representations

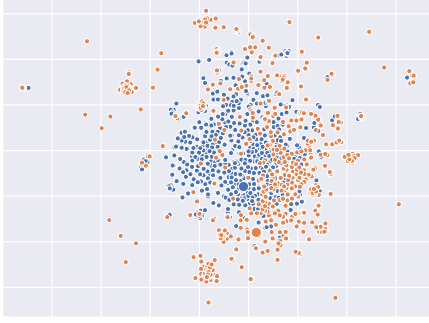


Figure 1: Head-tail entity pairs of relation “be an unincorporated community in” (in blue) and “be a small city in” (in red) sampled from our fact distribution model. The coordinates of the points are computed by t-sne (Maaten and Hinton, 2008) on the concatenation of head and tail embeddings⁸. The two **larger** blue and red points indicate the embeddings of these two relations.

the capacity of a single fixed-size representation is limited — some details about the fact distribution is lost during embedding; (2) directly comparing distributions yields a better interpretability — you can not know about how two relations are different given two relation embeddings, but our model helps you study the detailed differences between probabilities on every entity pair. Figure 1 provides an example. Although the two relations talk about the same topic, they have different meanings. TransE embeds them as vectors the closest to each other, while our model can capture the distinction between the distributions corresponds to the two relations, which could be directly noticed from the figure.

4 Dataset Construction

We show the statistics of the dataset we use in Table 3, and the construction procedures will be introduced in this section.

4.1 Wikidata

In Wikidata (Vrandečić and Krötzsch, 2014), facts can be described as (Head item/property, *Property*, Tail item/property). To construct a dataset suitable for our task, we only consider the facts whose head

⁸Embeddings used in this graph are from a trained TransE model.

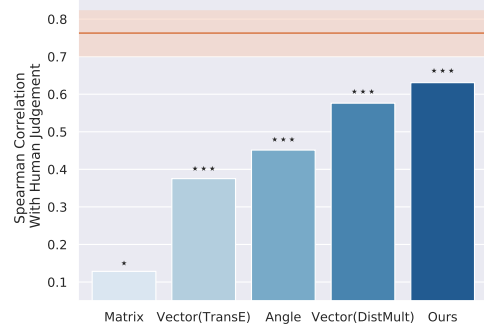


Figure 2: Spearman correlations between human judgment and models’ outputs. The inter-subject correlation is also shown as a horizontal line with its standard deviation as an error band. Our model shows the strongest positive correlation with human judgment, and, in other words, the smallest margin with human inter-subject agreement. Significance: ***/**/* := $p < .001/.01/.05$.

entity and tail entity are both items. We first choose the most common 202 relations and 120000 entities from Wikidata as our initial data. Considering that the facts containing the two most frequently appearing relations (*P2860: cites*, and *P31: instance of*) occupy half of the initial data, we drop the two relations to downsize the dataset and make the dataset more balanced. Finally, we keep the triples whose head and tail both come from the selected 120000 entities as well as its relation comes from the remaining 200 relations.

4.2 ReVerb Extractions

ReVerb (Fader et al., 2011) is a program that automatically identifies and extracts binary relationships from English sentences. We use the extractions from running ReVerb on Wikipedia⁹. We only keep the relations appear more than 10 times and their corresponding triples to construct our dataset.

4.3 FB15K and TACRED

FB15K (Bordes et al., 2013) is a subset of freebase. TACRED (Zhang et al., 2017) is a large supervised relation extraction dataset obtained via crowdsourcing. We directly use these two dataset, no extra processing steps were applied.

⁹<http://reverb.cs.washington.edu/>

5 Human Judgments

Following Miller and Charles (1991); Resnik (1999) and the vast amount of previous work on semantic similarity, we ask nine undergraduate subjects to assess the similarity of 360 pairs of relations from a subset of Wikidata (Vrandečić and Krötzsch, 2014)¹⁰ that are chosen to cover from high to low levels of similarity. In our experiment, subjects were asked to rate an integer similarity score from 0 (no similarity) to 4 (perfectly the same)¹¹ for each pair. The inter-subject correlation, estimated by leaving-one-out method (Weiss and Kulikowski, 1991), is $r = 0.763$, standard deviation = 0.060. This important reference value (marked in Figure 2) could be seen as the highest expected performance for machines (Resnik, 1999).

To get baselines for comparison, we consider other possible methods to define similarity functions, as shown in Table 2. We compute the correlation between these methods and human judgment scores. As the models we have chosen are the ones work best in knowledge base completion, we do expect the similarity quantification approaches based on them could measure some degree of similarity. As shown in Figure 2, the three baseline models could achieve moderate (0.1–0.5) positive correlation. On the other hand, our model shows a stronger correlation (0.63) with human judgment, indicating that considering the probability over whole entity pair space helps to gain a similarity closer to human judgments. These results provide evidence for our claim raised in §3.2.

6 Redundant Relation Removal

Open IE extracts concise token patterns from plain text to represent various relations between entities, e.g., (Mark Twain, *was born in*, Florida). As Open IE is significant for constructing KBs, many effective extractors have been proposed to extract triples, such as Text-Runner (Yates et al., 2007), ReVerb (Fader et al., 2011), and Stanford Open IE (Angeli et al., 2015). However, these extractors only yield relation patterns between entities, without aggregating and clustering their results. Accordingly, there are a fair amount of redundant relation patterns after extracting those relation patterns. Furthermore, the redundant patterns lead to

¹⁰Wikidata provides detailed descriptions to properties (relations), which could help subjects understand the relations better.

¹¹The detailed instruction is attached in the Appendix F.

Triple Set	$ \mathcal{R} $	$ \mathcal{E} $	#Fact	Section
Wikidata	188	112,946	426,067	§5 and §6.1
ReVerb Extractions	3,736	194,556	266,645	§6.2
FB15K	1,345	14,951	483,142	§7.1 and §8
TACRED	42	29,943	68,124	§7.2 and §9

Table 3: Statistics of the triple sets used in this paper.

some redundant relations in KBs.

Recently, some efforts are devoted to Open Relation Extraction (Open RE) (Lin and Pantel, 2001; Yao et al., 2011; Marcheggiani and Titov, 2016; ElSahar et al., 2017), aiming to cluster relation patterns into several relation types instead of redundant relation patterns. Whenas, these Open RE methods adopt distantly supervised labels as golden relation types, suffering from both false positive and false negative problems on the one hand. On the other hand, these methods still rely on the conventional similarity metrics mentioned above.

In this section, we will show that our defined similarity quantification could help Open IE by identifying redundant relations. To be specific, we set a toy experiment to remove redundant relations in KBs for a preliminary comparison (§6.1). Then, we evaluate our model and baselines on the real-world dataset extracted by Open IE methods (§6.2). Considering the existing evaluation metric for Open IE and Open RE rely on either labor-intensive annotations or distantly supervised annotations, we propose a metric approximating recall and precision evaluation based on operable human annotations for balancing both efficiency and accuracy.

6.1 Toy Experiment

In this subsection, we propose a toy environment to verify our similarity-based method. Specifically, we construct a dataset from Wikidata¹² and implement Chinese restaurant process¹³ to split every relation in the dataset into several sub-relations. Then, we filter out those sub-relations appearing less than 50 times to eventually get 1165 relations. All these split relations are regarded as different ones during training, and then different relation similarity metrics are adopted to merge those sub-relations into one relation. As Figure 2 shown that the matrices-based approach is less effective than other approaches, we leave this approach out of this experiment. The results are shown in Table 4.

¹²The construction procedure is shown in §4.1.

¹³Chinese restaurant process is shown in Appendix B.

Method	P	R	F_1
Vectors (TransE)	0.28	0.14	0.18
Vectors (DistMult)	0.44	0.41	0.42
Angles	0.48	0.43	0.45
Ours	0.65	0.50	0.57

Table 4: The experiment results on the toy dataset show that our metric based on probability distribution significantly outperforms other relation similarity metrics.

6.2 Real World Experiment

In this subsection, we evaluate various relation similarity metrics on the real-world Open IE patterns. The dataset are constructed by ReVerb. Different patterns will be regarded as different relations during training, and we also adopt various relation similarity metrics to merge similar relation patterns. Because it is nearly impossible to annotate all pattern pairs for their merging or not, meanwhile it is also inappropriate to take distantly supervised annotations as golden results. Hence, we propose a novel metric approximating recall and precision evaluation based on minimal human annotations for evaluation in this experiment.

Approximating Recall and Precision

Recall Recall is defined as the yielding fraction of true positive instances over the total amount of real positive¹⁴ instances. However, we do not have annotations about which pairs of relations are synonymous. Crowdsourcing is a method to obtain a large number of high-quality annotations. Nevertheless, applying crowdsourcing is not trivial in our settings, because it is intractable to enumerate all synonymous pairs in the large space of relation (pattern) pairs $\mathcal{O}(|\mathcal{R}|^2)$ in Open IE. A promising method is to use rejection sampling by uniform sampling from the whole space, and only keep the synonymous ones judged by crowdworkers. However, this is not practical either, as the synonymous pairs are sparse in the whole space, resulting in low efficiency. Fortunately, we could use normalized importance sampling as an alternative to get an unbiased estimation of recall.

Theorem 1.¹⁵ Suppose every sample $x \in X$ has a label $f(x) \in \{0, 1\}$, and the model to be evaluated also gives its prediction $\hat{f}(x) \in \{0, 1\}$. The recall can be written as

$$Recall = \mathbb{E}_{x \sim U} \mathbb{I}[\hat{f}(x) = 1], \quad (9)$$

where U is the uniform distribution over all samples with $f(x) = 1$. If we have a proposal distribu-

¹⁴Often called relevant in information retrieval field.

¹⁵See proof in Appendix A

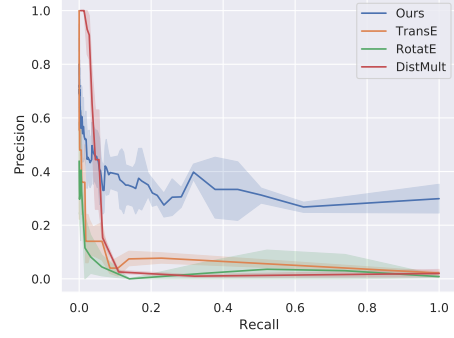


Figure 3: Precision-recall curve on Open IE task comparing our similarity function with vector-based and angle-based similarity. Error bar represents 95% confidential interval. Bootstrapping is used to calculate the confidential interval.

tion $q(x)$ satisfying $\forall x, f(x) = 1 \wedge \hat{f}(x) = 1 \Rightarrow q(x) \neq 0$, we get an unbiased estimation of recall:

$$Recall \approx \sum_{i=1}^n \mathbb{I}[\hat{f}(x_i) = 1] \hat{w}_i, \quad (10)$$

where \hat{w}_i is a normalized version of $w_i = \frac{\mathbb{I}[f(x_i)=1]}{\tilde{q}(x_i)}$, where \tilde{q} is the unnormalized version of q , and $\{x_i\}_{i=1}^n$ are i.i.d. drawn from $q(x)$.

Precision Similar to equation (9), we can write the expectation form of precision:

$$Precision = \mathbb{E}_{x \sim U'} \mathbb{I}[f(x) = 1], \quad (11)$$

where U' is the uniform distribution over all samples with $\hat{f}(x) = 1$. As these samples could be found out by performing models on it. We can simply approximate precision by Monte Carlo Sampling:

$$Precision \approx \frac{1}{n} \sum_{i=1}^n \mathbb{I}[f(x_i) = 1], \quad (12)$$

where $\{x_i\}_{i=1}^n \stackrel{i.i.d.}{\sim} U'$.

In our setting, $x = (r_1, r_2) \in \mathcal{R} \times \mathcal{R}$, $f(x) = 1$ means r_1 and r_2 are the same relations, $\hat{f}(x) = 1$ means $S(r_1, r_2)$ is larger than a threshold λ .

Results

The results on the ReVerb Extractions dataset that we constructed are described in Figure 3. To approximate recall, we use the similarity scores as the proposal distribution \tilde{q} . 500 relation pairs are then drawn from \tilde{q} . To approximate precision, we set thresholds at equal intervals. At each threshold, we uniformly sample 50 to 100 relation pairs whose similarity score given by the model is larger than the threshold. We ask 15 undergraduates to judge

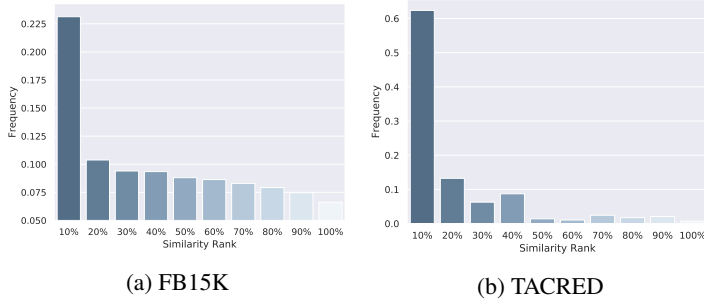


Figure 4: Similarity rank distributions of distracting relations on different tasks and datasets. Most of the distracting relations have top similarity rank. Distracting relations are, as defined previously, the relations have a higher rank in the relation classification result than the ground truth.

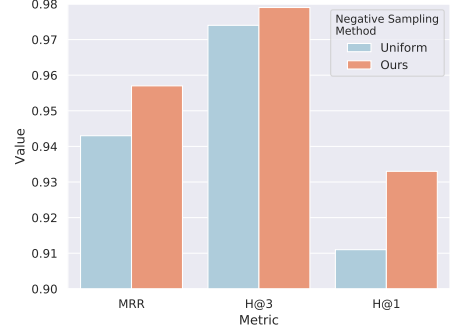


Figure 5: Improvement of using similarity in a heuristic method for negative sampling. MRR denotes the mean reciprocal rank.

whether two relations in a relation pair have the same meaning. A relation pair is viewed valid only if 8 of the annotators annotate it as valid. We use the annotations to approximate recall and precision with equation (10) and equation (12). Apart from the confidential interval of precision shown in the figure, the largest 95% confidential interval among thresholds for recall is 0.04¹⁶. From the result, we could see that our model performs much better than other models' similarity by a very large margin.

7 Error Analysis for Relational Classification

In this section, we consider two kinds of relational classification tasks: (1) relation prediction and (2) relation extraction. Relation prediction aims at predicting the relationship between entities with a given set of triples as training data; while relation extraction aims at extracting the relationship between two entities in a sentence.

7.1 Relation Prediction

We hope to design a simple and clear experiment setup to conduct error analysis for relational prediction. Therefore, we consider a typical method TransE (Bordes et al., 2013) as the subject as well as FB15K (Bordes et al., 2013) as the dataset. TransE embeds entities and relations as vectors, and train these embeddings by minimizing

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{D}} [d(\mathbf{h} + \mathbf{r}, \mathbf{t}) - d(\mathbf{h}' + \mathbf{r}', \mathbf{t}') + \gamma]_+, \quad (13)$$

¹⁶The figure is shown in Figure 6

where \mathcal{D} is the set of training triples, $d(\cdot, \cdot)$ is the distance function, (h', r', t') ¹⁷ is a negative sample with one element different from (h, r, t) uniformly sampled from $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and γ is the margin.

During testing, for each entity pair (h, t) , TransE rank relations according to $d(\mathbf{h} + \mathbf{r}, \mathbf{t})$. For each (h, r, t) in the test set, we call the relations with higher rank scores than r **distracting relations**. We then compare the similarity between the golden relation and **distracting relations**. Note that some entity pairs could correspond to more than one relations, in which case we just do not see them as distracting relations.

7.2 Relation Extraction

For relation extraction, we consider the supervised relation extraction setting and TACRED dataset (Zhang et al., 2017). As for the subject model, we use the best model on TACRED dataset — position-aware neural sequence model. This method first passes the sentence into an LSTM and then calculate an attention sum of the hidden states in the LSTM by taking positional features into account. This simple and effective method achieves the best in TACRED dataset.

7.3 Results

Figure 4 shows the distribution of similarity ranks of distracting relations of the above mentioned models' outputs on both relation prediction and relation extraction tasks. From Figures 4a and 4b, we could observe the most distracting relations are the most

¹⁷Note that only head and tail entities are changed in the original TransE when doing link prediction. But changing r' results in better performance when doing relation prediction.

	Model	P	R	F_1
Traditional	Patterns	86.9	23.2	36.6
	LR	73.5	49.9	59.4
Neural	CNN	75.6	47.5	58.3
	CNN-PE	70.3	54.2	61.2
	SDP-LSTM (Xu et al., 2015)	66.3	52.7	58.7
	LSTM	65.7	59.9	62.7
	PA-LSTM (Zhang et al., 2017)	65.7	64.5	65.1
Neural+Ours	PA-LSTM (Softmax-Margin Loss)	68.5	64.7	66.6

Table 5: Improvement of using similarity in softmax-margin loss.

similar ones, which corroborate our hypothesis that even the best models on these tasks still make mistakes among the most similar relations. This result also highlights the importance of a heuristic method for guiding models to pay more attention to the boundary between similar relations. We also try to do the negative sampling with relation type constraints, but we see no improvement compared with uniform sampling. The details of negative sampling with relation type constraints are presented in Appendix E.

8 Similarity and Negative Sampling

Based on the observation presented in §7.3, we find out that similar relations are often confusing for relation prediction models. Therefore, corrupted triples with similar relations can be used as high-quality negative samples.

For a given valid triple (h, r, t) , we corrupt the triple by substituting r with r' with the probability,

$$p = \frac{S(r, r')^{1/\alpha}}{\sum_{r'' \in \mathcal{R} \setminus \{r\}} S(r, r'')^{1/\alpha}}, \quad (14)$$

where α is the temperature of the exponential function, the bigger the α is, the flatter the probability distribution is. When the temperature approaches infinite, the sampling process reduces to uniform sampling.

In training, we set the initial temperature to a high level and gradually reduce the temperature. Intuitively, it enables the model to distinguish among those obviously different relations in the early stage and gives more and more confusing negative triples as the training processes to help the model distinguish the similar relations. This can be also viewed as a process of curriculum learning (Bengio et al., 2009), the data fed to the model gradually changes from simple negative triples to hard ones.

We perform relation prediction task on FB15K with TransE. Following Bordes et al. (2013), we use the "Filtered" setting protocol, i.e., filtering out

the corrupted triples that appear in the dataset. Our sampling method is shown to improve the model's performance, especially on Hit@1 (Figure 5). Training details are described in Appendix C.

9 Similarity and Softmax-Margin Loss

Similar to §8, we find out that relation extraction models often make wrong predictions on similar relations. In this section, we use similarity as an adaptive margin in softmax-margin loss to improve the performance of relation extraction models.

As shown in (Gimpel and Smith, 2010), Softmax-Margin Loss can be expressed as

$$\mathcal{L} = \sum_{i=1}^n -\theta^T f(x^{(i)}, r^{(i)}) + \log \sum_{r \in \mathcal{R}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, r) + \text{cost}(r^{(i)}, r)\}, \quad (15)$$

where $\mathcal{R}(x)$ denotes a structured output space for x , and $\langle x^{(i)}, r^{(i)} \rangle$ is i^{th} example in training data.

We can easily incorporate similarity into cost function $\text{cost}(r^{(i)}, r)$. In this task, we define the cost function as $\alpha S(r^{(i)}, r)$, where α is a hyperparameter.

Intuitively, we give a larger margin between similar relations, forcing the model to distinguish among them, and thus making the model perform better. We apply our method to Position-aware Attention LSTM (PA-LSTM) (Zhang et al., 2017), and Table 5 shows our method improves the performance of PA-LSTM. Training details are described in Appendix C.

10 Related Works

As many early works devoted to psychology and linguistics, especially those works exploring semantic similarity (Miller and Charles, 1991; Resnik, 1999), researchers have empirically found there are various different categorizations of semantic relations among words and contexts. For promoting research on these different semantic relations, Bejar et al. (1991) explicitly defining these relations and Miller (1995) further systematically organize rich semantic relations between words via a database. For identifying correlation and distinction between different semantic relations so as to support learning semantic similarity, various methods have attempted to measure relational similarity (Turney, 2005, 2006; Zhila et al., 2013; Pedersen, 2012; Rink and Harabagiu, 2012; Mikolov et al., 2013b,a).

With the ongoing development of information extraction and effective construction of KBs (Suchanek et al., 2007; Bollacker et al., 2008; Bizer et al., 2009), relations are further defined as various types of latent connections between objects more than semantic relations. These general relations play a core role in expressing relational facts in the real world. Hence, there are accordingly various methods proposed for discovering more relations and their facts, including open information extraction (Brin, 1998; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002; Banko et al., 2007; Zhu et al., 2009; Etzioni et al., 2011; Saha et al., 2017) and relation extraction (Riedel et al., 2013; Liu et al., 2013; Zeng et al., 2014; Santos et al., 2015; Zeng et al., 2015; Lin et al., 2016), and relation prediction (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015b,a; Xie et al., 2016).

For both semantic relations and general relations, identifying them is a crucial problem, requiring systems to provide a fine-grained relation similarity metric. However, the existing methods suffer from sparse data, which makes it difficult to achieve an effective and stable similarity metric. Motivated by this, we propose to measure relation similarity by leveraging their fact distribution so that we can identify nuances between similar relations, and merge those distant surface forms of the same relations, benefitting the tasks mentioned above.

11 Conclusion and Future Work

In this paper, we introduce an effective method to quantify the relation similarity and provide analysis and a survey of applications. We note that there are a wide range of future directions: (1) human prior knowledge could be incorporated into the similarity quantification; (2) similarity between relations could also be considered in multi-modal settings, e.g., extracting relations from images, videos, or even from audios; (3) by analyzing the distributions corresponding to different relations, one can also find some “meta-relations” between relations, such as hypernymy and hyponymy.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC No. 61572273, 61532010), the National Key Research and Development Program of China (No. 2018YFB1004503). Chen and Zhu is supported by Tsinghua University Initiative Scientific Research Program, and Chen is

also supported by DCST Student Academic Training Program. Han is also supported by 2018 Tencent Rhino-Bird Elite Training Program.

References

- Eugene Agichtein and Luis Gravano. 2000. [Snowball: Extracting relations from large plain-text collections](#). In *Proceedings of JCDL*, pages 85–94.
- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. [Leveraging linguistic structure for open domain information extraction](#). In *Proceedings of ACL*, pages 344–354.
- Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. [Open information extraction from the web](#). In *Proceedings of IJCAI*, pages 2670–2676.
- Isaac I Bejar, Roger Chaffin, and Susan E Embretson. 1991. [Cognitive and psychometric analysis of analogical problem solving](#).
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *Proceedings of ICML*, pages 41–48.
- Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. [Dbpedia-a crystallization point for the web of data](#). *Web Semantics: science, services and agents on the world wide web*, 7:154–165.
- 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 SIGMOD*, pages 1247–1250.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. [Translating embeddings for modeling multi-relational data](#). In *Proceedings of NIPS*, pages 2787–2795.
- Sergey Brin. 1998. [Extracting patterns and relations from the world wide web](#). In *Proceedings of WWW*, pages 172–183.
- Hady ElSahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. [Unsupervised open relation extraction](#). In *Proceedings of ESWC*, pages 12–16.
- Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. [Open information extraction: the second generation](#). In *Proceedings of IJCAI*, pages 3–10.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. [Identifying relations for open information extraction](#). In *Proceedings of EMNLP*, pages 1535–1545.

- Kevin Gimpel and Noah A Smith. 2010. [Softmax-margin crfs: Training log-linear models with cost functions](#). In *Proceedings of NAACL*, pages 733–736.
- Diederik P Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *arXiv preprint arXiv:1412.6980*.
- Dekang Lin and Patrick Pantel. 2001. [Dirt@ sbt@ discovery of inference rules from text](#). In *Proceedings of SIGKDDs*, pages 323–328.
- Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015a. [Modeling relation paths for representation learning of knowledge bases](#). In *Proceedings of EMNLP*, pages 705–714.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015b. [Learning entity and relation embeddings for knowledge graph completion](#). In *Proceedings of AAAI*, pages 2181–2187.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. [Neural relation extraction with selective attention over instances](#). In *Proceedings of ACL*, pages 2124–2133.
- ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. 2013. [Convolution neural network for relation extraction](#). In *Proceedings of ICDM*, pages 231–242.
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-sne](#). *JMLR*, 9:2579–2605.
- Diego Marcheggiani and Ivan Titov. 2016. [Discrete-state variational autoencoders for joint discovery and factorization of relations](#). *TACL*, 4:231–244.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. [Efficient estimation of word representations in vector space](#). In *Proceedings of ICLR*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. [Distributed representations of words and phrases and their compositionality](#). In *Proceedings of NIPS*, pages 3111–3119.
- George A Miller. 1995. [Wordnet: a lexical database for english](#). *Communications of the ACM*, 38:39–41.
- George A Miller and Walter G Charles. 1991. [Contextual correlates of semantic similarity](#). *Language and cognitive processes*, 6:1–28.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. [A three-way model for collective learning on multi-relational data](#). In *Proceedings of ICML*, pages 809–816.
- Art B. Owen. 2013. *Monte Carlo theory, methods and examples*.
- Ted Pedersen. 2012. [Duluth: Measuring degrees of relational similarity with the gloss vector measure of semantic relatedness](#). In *Proceedings of SemEval 2012*, pages 497–501.
- Deepak Ravichandran and Eduard Hovy. 2002. [Learning surface text patterns for a question answering system](#). In *Proceedings of ACL*, pages 41–47.
- Philip Resnik. 1999. [Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language](#). *Journal of artificial intelligence research*, 11:95–130.
- Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. [Relation extraction with matrix factorization and universal schemas](#). In *Proceedings of NAACL*, pages 74–84.
- Bryan Rink and Sanda Harabagiu. 2012. [Utd: Determining relational similarity using lexical patterns](#). In *Proceedings of SemEval 2012*, pages 413–418.
- Swarnadeep Saha, Harinder Pal, et al. 2017. [Bootstrapping for numerical open ie](#). In *Proceedings of ACL*, volume 2, pages 317–323.
- Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. [Classifying relations by ranking with convolutional neural networks](#). In *Proceedings of ACL-IJCNLP*, pages 626–634.
- Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. [Yago: a core of semantic knowledge](#). In *Proceedings of WWW*, pages 697–706.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [Rotate: Knowledge graph embedding by relational rotation in complex space](#). In *Proceedings of ICLR*.
- Peter D Turney. 2005. [Measuring semantic similarity by latent relational analysis](#). In *Proceedings of IJCAI*, pages 1136–1141.
- Peter D Turney. 2006. [Similarity of semantic relations](#). *Computational Linguistics*, 32:379–416.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wikidata: a free collaborative knowledgebase](#). *Communications of the ACM*, 57:78–85.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. [Knowledge graph embedding by translating on hyperplanes](#). In *Proceedings of AAAI*, pages 1112–1119.
- Sholom M Weiss and Casimir A Kulikowski. 1991. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*.
- Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. [Representation learning of knowledge graphs with hierarchical types](#). In *Proceedings of IJCAI*, pages 2965–2971.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. [Classifying relations via long short term memory networks along shortest dependency paths](#). In *Proceedings of EMNLP*, pages 1785–1794.

- 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 *Proceedings of ICLR*.
- Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. [Structured relation discovery using generative models](#). In *Proceedings of EMNLP*, pages 1456–1466.
- Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. [Textrunner: open information extraction on the web](#). In *Proceedings of NAACL*, pages 25–26.
- Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. [Distant supervision for relation extraction via piecewise convolutional neural networks](#). In *Proceedings of EMNLP*, pages 1753–1762.
- 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.
- Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. [Position-aware attention and supervised data improve slot filling](#). In *Proceedings of EMNLP*, pages 35–45.
- Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. [Combining heterogeneous models for measuring relational similarity](#). In *Proceedings of NAACL*, pages 1000–1009.
- Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. [Statsnowball: a statistical approach to extracting entity relationships](#). In *Proceedings of WWW*, pages 101–110.

A Proofs to theorems in the paper

Proof.

$$\begin{aligned}
Recall &= \frac{\sum_x \mathbb{I}[f(x) = 1 \wedge \hat{f}(x) = 1]}{\sum_x \mathbb{I}[f(x) = 1]} \\
&= \sum_x \frac{\mathbb{I}[f(x) = 1 \wedge \hat{f}(x) = 1]}{\sum_{x'} \mathbb{I}[f(x') = 1]} \\
&= \sum_x \frac{\mathbb{I}[f(x) = 1] \mathbb{I}[\hat{f}(x) = 1]}{\sum_{x'} \mathbb{I}[f(x') = 1]} \\
&= \sum_x \frac{\mathbb{I}[f(x) = 1]}{\sum_{x'} \mathbb{I}[f(x') = 1]} \mathbb{I}[\hat{f}(x) = 1] \\
&= \sum_x P_U(x) \mathbb{I}[\hat{f}(x) = 1] \\
&= \mathbb{E}_{x \sim U} \mathbb{I}[\hat{f}(x) = 1]
\end{aligned} \tag{16}$$

If we have a proposal distribution $q(x)$ satisfying $\forall x, f(x) = 1 \wedge \hat{f}(x) = 1 \Rightarrow q(x) \neq 0$, then equation (16) can be further written as

$$Recall = \mathbb{E}_{x \sim q} \mathbb{I}[\hat{f}(x) = 1] \frac{P_U(x)}{q(x)} \tag{17}$$

Sometimes, it's hard for us to compute normalized probability q . To tackle this problem, consider self-normalized importance sampling as an unbiased estimation (Owen, 2013),

$$\begin{aligned}
&\mathbb{E}_{x \sim q} \mathbb{I}[\hat{f}(x) = 1] \frac{P_U(x)}{q(x)} \\
&\approx \frac{\sum_{i=1}^n \mathbb{I}[\hat{f}(x_i) = 1] P_U(x_i) / q(x_i)}{\sum_{i=1}^n P_U(x_i) / q(x_i)} \\
&= \frac{\sum_{i=1}^n \mathbb{I}[\hat{f}(x_i) = 1] w_i}{\sum_{i=1}^n w_i} \quad (w_i = \frac{\mathbb{I}[f(x_i) = 1]}{\tilde{q}(x_i)}) \\
&= \sum_{i=1}^n \mathbb{I}[\hat{f}(x_i) = 1] \hat{w}_i,
\end{aligned} \tag{18}$$

where \hat{w}_i is the normalized version of w . \square

B Chinese Restaurant Process

Specifically, for a relation r with currently m sub-relations, we turn it to a new sub-relation with probability

$$p = \frac{\alpha}{\alpha + n + 1} \tag{19}$$

or to the k^{th} existing sub-relation with probability

$$p = \frac{n_k}{\alpha + n + 1} \tag{20}$$

where n_k is the size of k^{th} existing sub-relation, n is the sum of the number of all sub-relationships of r , and α is a hyperparameter, in which case we use $\alpha = 1$.

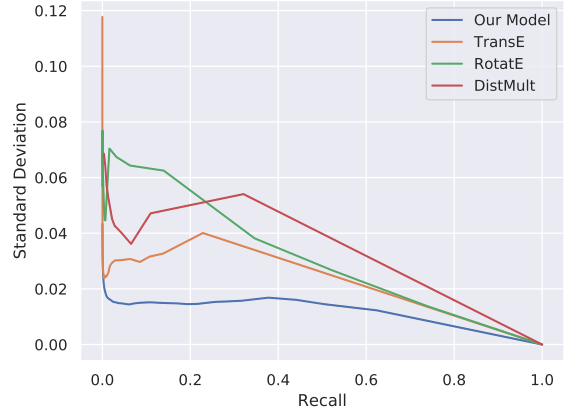


Figure 6: The recall standard deviation of different models.

C Training Details

In Wikidata and ReVerb Extractions dataset, we manually split a validation set, assuring every entity and relation appears in validation set also appears in training set. While minimizing loss on the training set, we observe the loss on the validation set and stop training as validation loss stops to decrease. Before training our model on any dataset, we use the entity embeddings and relation embeddings produced by TransE on the dataset as the pretrained embeddings for our model.

C.1 Training Details on Negative Sampling

The sampling is launched with an initial temperature of 8192. The temperature drops to half every 200 epochs and remains stable once it hits 16. Optimization is performed using SGD, with a learning rate of $1e-3$.

C.2 Training Details on Softmax-Margin Loss

The sampling is launching with an initial temperature of 64. The temperature drops by 20% per epoch, and remains stable once it hits 16. The alpha we use is 9. Optimization is performed using SGD, with a learning rate of 1.

D Recall Standard Deviation

As is shown in Figure 6, the max recall standard deviation for our model is 0.4, and 0.11 for TransE.

E Negative Sampling with Relation Type Constraints

In FB15K, if two relations have same prefix, we regard them as belonging to a same type, e.g., both `/film/film/starring./film/performance/actor` and

/film/actor/film./film/performance/film have prefix *film*, they belong to same type. Similar to what is mentioned in §8, we expect the model first to learn to distinguish among obviously different relations, and gradually learn to distinguish similar relations. Therefore, we conduct negative sampling with relation type constraints in two ways.

E.1 Add Up Two Uniform Distribution

For each triple (h, r, t) , we have two uniform distribution U_{all} and U_{type} . U_{all} is the uniform distribution over all the relations except for those appear with (h, t) in the knowledge base, and U_{type} is the uniform distribution over the relations of the same type as r . When corrupting the triple, we sample r' from the distribution:

$$U = \alpha U_{all} + (1 - \alpha) U_{type}, \quad (21)$$

where α is a hyperparameter. We set α to 1 at the beginning of training, and every k epochs, α will be multiplied by decrease rate γ . We do grid search for $k \in \{50, 70, 100\}$ and $\gamma \in \{0.9, 0.95, 0.98\}$, but no improvement is observed.

E.2 Add Weight

We speculate that the unsatisfactory result produced by adding up two uniform distribution is because that for those types with few relations in it, a small change of α will result in a significant change in U . Therefore, when sampling a negative r' , we add weights to relations that are of the same type as r instead. Concretely, we substitute r with r' with probability p , which can be calculated as:

$$p = \begin{cases} \frac{1+\epsilon}{N} & r' \in \mathcal{T}(r) \\ \frac{1}{N} & \text{otherwise} \end{cases} \quad (22)$$

where $\mathcal{T}(r)$ denotes all the relations that are the same type as r , ϵ is a hyperparameter and N is a normalizing constant. We set ϵ to 0 at the beginning of training, and every k epochs, ϵ will increase by γ . We do grid search for $k \in \{50, 70, 100\}$ and $\gamma \in 0.5, 1$, still no improvement is observed.

F Wikidata annotation guidance

We show the guidance provided for the annotators here.

- A pair of relations should be marked as **4** points if the two relations are only two different expressions for a certain meaning.

Example: (study at, be educated at)

- A pair of relations should be marked as **3** points if the two relations are describing a **same topic**, and the entities that the two relations connect are of **same type** respectively.

Example: (be the director of, be the screenwriter of), both relations relate to movie, and the types of the entities they connect are both (person, movie).

- A pair of relations should be marked as **2** points if the two relations are describing a **same topic**, but the entities that the two relations connect are of **different type** respectively.

Example: (be headquartered in, be founded in), both relations relate to organization, but the types of the entities they connect are different, i.e., (company, location) and (company, time)

- A pair of relations should be marked as **1** points if the two relations do not meet the conditions above but still have semantic relation.

Example: (be the developer of, be the employer of)

- A pair of relations should be marked as **0** points if the two relations do not have any connection.

Example: (be a railway station locates in, be published in)