Learning to Explain Entity Relationships by Pairwise Ranking with Convolutional Neural Networks

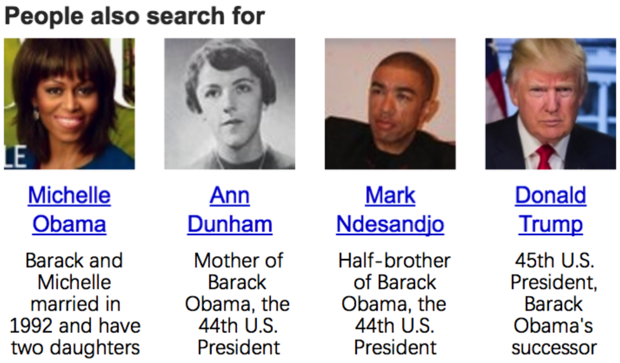
Jizhou Huang[†,‡*], Wei Zhang[‡], Shiqi Zhao[‡], Shiqiang Ding[‡], Haifeng Wang[‡]
[†]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[‡]Baidu Inc., Beijing, China
{huangjizhou01, zhangwei32, zhaoshiqi, dingshiqiang01, wanghaifeng}@baidu.com

Abstract

Providing a plausible explanation for the relationship between two related entities is an important task in some applications of knowledge graphs, such as in search engines. However, most existing methods require a large number of manually labeled training data, which cannot be applied in large-scale knowledge graphs due to the expensive data annotation. In addition, these methods typically rely on costly handcrafted features. In this paper, we propose an effective pairwise ranking model by leveraging clickthrough data of a Web search engine to address these two problems. We first construct large-scale training data by leveraging the query-title pairs derived from clickthrough data of a Web search engine. Then, we build a pairwise ranking model which employs a convolutional neural network to automatically learn relevant features. The proposed model can be easily trained with backpropagation to perform the ranking task. The experiments show that our method significantly outperforms several strong baselines.

1 Introduction

Due to its heterogeneity, semantic richness, and large scale, knowledge graph has been widely used in search applications, such as search results enrichment with semantic information [Singhal, 2012] and entity recommendation [Blanco *et al.*, 2013; Yu *et al.*, 2014]. Presenting evidence on why two entities are related in a knowledge graph can help in building and promoting trust between the users and the entity recommendation system [Voskarides *et al.*, 2015]. Figure 1 shows an example of Baidu[1] Web search engine's entity recommendations for the query "Obama". In this example, a short sentence is presented under each entity, e.g., "Barack and Michelle married in 1992 and have two daughters", providing users an explanation of the relationship between the recommended entity "Michelle Obama" and the queried entity "Obama".[2] Presenting related entities with plausible explanations can significantly increase the understandability of the recommendations and user engagement [Fang *et al.*, 2011; Huang *et al.*, 2016]. It is therefore important to find descriptive sentences that explain the relationship between two entities.



People also search for

Michelle Obama
Barack and Michelle married in 1992 and have two daughters

Ann Dunham
Mother of Barack Obama, the 44th U.S. President

Mark Ndesandjo
Half-brother of Barack Obama, the 44th U.S. President

Donald Trump
45th U.S. President, Barack Obama's successor

Figure 1: An example of entity recommendation with entity relationship explanation.

Although there is a growing interest in predicting relations between entities, e.g., [Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015; Ji *et al.*, 2016], the relation types defined by the knowledge graph, such as "Spouse" and "Offspring", can hardly be directly used as descriptions in the aforementioned application in search engines, which are not explanatory or informative enough. To better explain why two entities are related in a knowledge graph, it is important to provide more detailed evidence about the relationship between them. To the best of our knowledge, the problem of automatically finding a descriptive sentence that explains a given relationship of two entities has not been well addressed.

Although using a template-based method to match or generate a descriptive sentence for this task is a straightforward method, it has two major limitations. First, it is usually required to manually label a number of seeds or candidate templates for each relationship. Therefore, it can hardly be applied in large-scale knowledge graphs due to the expensive annotation. Second, this method can achieve high precision but low recall due to strict or limited templates.

Recently, Voskarides *et al.* [2015] first studied the problem of explaining relationships between entities. They modeled

[*] Corresponding author.

[1] https://www.baidu.com/

[2] We translate the example from Chinese to English for the sake of understanding.

this task as a ranking problem, and proposed a learning to rank (LTR) method with a rich set of handcrafted features to rank the sentences within Wikipedia[3] articles according to how well they describe the relationship between the entities. Although the method achieved promising results in the experiments, it has two major drawbacks when applied to large-scale, real-world problems.

First, large-scale training data is essential for building ranking models based on supervised machine learning techniques. To train the ranking model, a total of 5,689 sentences for 1,476 entity pairs are manually labeled as training and test data in [Voskarides *et al.*, 2015]. However, it gets too expensive if we want to generate a much larger scale training dataset, which is essential in training neural network models as we do in this paper.

Second, this method employs elaborately designed features. However, since errors exist inevitably in the feature extraction process, the use of handcrafted features leads to error propagation or accumulation [Zeng *et al.*, 2015]. With the recent revival of interest in neural networks, many researchers have tried to use neural networks to automatically learn features. It has been shown that this methd achieves significant improvements compared with manually designed features in several tasks, e.g., [Socher *et al.*, 2012; Kim, 2014; Zeng *et al.*, 2015; Santos *et al.*, 2015]. Inspired by these studies, we also employ a convolutional neural network (CNN) to automatically learn relevant features in our task.

To address the two problems above, this paper proposes an effective pairwise ranking model by leveraging clickthrough data. We first automatically extract large-scale training data by leveraging the query-title pairs derived from clickthrough data of a Web search engine. No manually labeled training data are needed. More specifically, we consider a special kind of query which consists of a relation triplet (an entity pair and their relationship), and view the corresponding clicked titles of the query as potential descriptions of the relationship between two entities mentioned in the query. We then build a pairwise ranking model that employs a CNN to automatically learn relevant features, in which no handcrafted features are needed. Finally, we use the trained model to rank the sentences according to how well they describe the relationship between the entities. The experimental results on a manually constructed test set show that our method significantly outperforms several strong baselines.

2 Related Work

Previous work that is the closest to our task is selecting sentences that describe a particular relationship of two entities in [Voskarides *et al.*, 2015]. However, there are some significant differences. First, we employ a CNN to automatically learn relevant features directly from the training examples, rather than use handcrafted features. Second, we propose an effective pairwise ranking model to rank the sentences instead of using a pointwise ranking model.

There are also some similar studies. Blanco and Zaragoza [2010] proposed to extract and rank sentences as contextual information to help users understand the relevance and relationship between an entity and a query. The major distinction between our work and this study is that our method focuses on explaining the relationship between two entities, rather than between an entity and a query. Fang *et al.* [2011] proposed to produce a ranked list of relationships that describe how a pair of entities are related based on a knowledge graph. For example, for “Angelina Jolie” and “Brad Pitt”, it generates “Spouse” and “Co-starring” as the explanations of their connections. Banko *et al.* [2007] and Fader *et al.* [2011] proposed to extract a large set of relational tuples using Open Information Extraction (IE) paradigm. Our work is different in that we extract sentences that are eligible for explaining the relationship of an entity pair, rather than recognize the types of the relationships.

3 Methodology

We study the problem of explaining relationships between entities by leveraging clickthrough data, and model this task as a sentence ranking problem. More specifically, given a relation triplet q_s consisted of a pair of entities e_h and e_t, and a relationship r_k between them, i.e., $q_s = (e_h, r_k, e_t)$, our task is to extract a set of candidate sentences $S = \{s_1, s_2, \cdots, s_n\}$ that contain e_h and e_t, and learn a ranking function to rank the sentences according to how well they describe the relationship r_k between e_h and e_t. In this paper, we focus on the ranking problem, and the candidate sentences are extracted using the keyword based retrieval method proposed in [Voskarides *et al.*, 2015]. In what follows, we first introduce the strategy we used to acquire large amounts of data for training ranking models. Then, we present our pairwise ranking model that employs a CNN to automatically learn relevant features, which is referred to as PR-CNN hereafter.

3.1 Training Data Acquisition

The acquisition of training data in the task of explaining relationships between entity pairs is crucial. Since publicly available datasets are quite limited, previous work [Voskarides *et al.*, 2015] manually labeled the training data by asking human assessors to assess the quality of each candidate description. However, this method is expensive and limited in quantity. Therefore, the training and evaluation are both restricted to a small number of labeled instances, making it difficult to meet the demand of large-scale, real-world applications. To alleviate this problem, we propose to acquire large amounts of training data by leveraging clickthrough data of a Web search engine.

The motivation of our approach is as follows. Given a query q_s that consists of an entity relation triplet (e_h, r_k, e_t), if a user clicks on a document title t that contains both entities e_h and e_t of the triplet when searching for q_s, then it is likely that the title t is a description of the entity relation triplet. Similar assumptions have been widely used in previous studies of search engines. For example, Zhao *et al.* [2010] proposed to extract paraphrases using the query-title pairs derived from clickthrough data of a Web search engine, based on an assumption that a query and its corresponding clicked document titles may mean the same thing. Gao *et al.* [2010]

[3] https://www.wikipedia.org/

Title		# clicks
t1	*Andy Lau* announces his marriage with *Carol Chu*	39
t2	*Andy Lau* and *Carol Chu* married for over two years	23
t3	*Andy Lau* confirms secret wedding to *Carol Chu*	10
t4	*Andy Lau* admitted his love relationship with *Carol Chu*	5
t5	How long has *Carol Chu* been waiting for *Andy Lau*	1
t6	Classic pictures of *Andy Lau*'s wife *Carol Chu*	0
t7	Biography of *Andy Lau*'s wife *Carol Chu*	0

Table 1: Aggregated clicks of titles for a query "*Andy Lau*'s wife *Carol Chu*" (e_h = *Andy Lau*, r_k = wife, e_t = *Carol Chu*).

proposed to learn the translation probability between phrases for improving retrieval effectiveness by using the query-title aligned corpus of a Web search engine, based on an assumption that a query is parallel to the titles of documents clicked on for that query. Huang *et al.* [2016] also made the same assumption and constructed large-scale monolingual parallel data of query-title aligned pairs to train machine translation models for a special application of sentence compression.

Furthermore, when aggregating all the clicked titles for a given query q_s, we assume that the titles get more clicks are better descriptions of the relation triplet than that get fewer, i.e., the former can better describe the relationship r_k between e_h and e_t mentioned in q_s than the latter. Table 1 shows several sampled titles with their aggregated clicks for a query "*Andy Lau*'s wife *Carol Chu*".[4] As can be seen from the examples, the titles with more clicks are obviously better descriptions of the given relation triplet than those with fewer clicks. When used for learning, valuable aspects and clues for ranking sentences can be learned from these titles. For example, compared with the words "pictures" and "biography" in t6 and t7, the phrases "announces his marriage with", "married for", "secret wedding", and "love relationship" in t1, t2, t3, and t4 are more commonly used formulas to describe the relationship "wife" for two people.

For a given query q_s, titles with more clicks might be more relevant to the query than the ones with no or relatively less clicks [Dou *et al.*, 2008]. Meanwhile, the query q_s consists of an entity relation triplet (e_h, r_k, e_t), and its corresponding clicked titles contain both entities e_h and e_t of the triplet. Therefore, titles that are more relevant to the query q_s can better describe the relationship r_k between e_h and e_t mentioned in q_s than the ones that are less relevant to q_s. For this reason, to construct the pairwise training data for learning the proposed ranking models, it is essential to extract pairwise relevance preferences from clickthrough data.

Joachims [2002] and Agichtein *et al.* [2006] showed that clickthrough data can be used to predict relative relevance preferences for the search results. Agichtein *et al.* [2006] further showed that clickthrough-only strategies could reach high precision, and recall could be improved quickly with more days of logs. Therefore, we used a six-months clickthrough data of a Web search engine to extract pairwise relevance preferences. Dou *et al.* [2008] studied the problem of using aggregated clickthrough data to learn Web search rankings, and showed that pairwise relevance preferences extracted from clickthrough data weakly correlate to human judgments on average, and a straightforward use of them as training examples can achieve a better ranking than using human judgments. We follow the method proposed in [Dou *et al.*, 2008] to extract training examples for learning ranking models. The basic idea is that, given a query, a pairwise training example is generated if one title receives more aggregated clicks than the other. Specifically, we use the following strategy to extract pairwise training examples.

Let $cdif(q_s, t_i, t_j) = click(q_s, t_i) - click(q_s, t_j)$*, where* $click(q_s, t)$ *is the aggregated click frequency of title* t *for query* q_s*, and* $cdif(q_s, t_i, t_j)$ *is click frequency difference of two titles* t_i *and* t_j *for query* q_s*. A relevance preference example* $rel(q_s, t_i) > rel(q_s, t_j)$ *is extracted for learning if* $cdif(q_s, t_i, t_j) > 0$.

3.2 Pairwise Ranking Model

To rank sentences, we present an effective pairwise ranking model that employs a CNN to automatically learn relevant features from the pairwise training examples acquired in Section 3.1. In the following we describe the details of our proposed model.

Network Architecture

Figure 2 shows the network architecture of the proposed pairwise ranking model. This network takes a triplet (q_s, t_i, t_j) as input that consists of a query $q_s = (e_h, r_k, e_t)$ and a pair of titles t_i and t_j, which are fed independently into three identical CNNs with shared architecture and parameters. In our experiments, each query q_s is formed by concatenating the triplet with the order $e_h \ r_k \ e_t$. The triplet characterizes the relative relevance score for t_i and t_j to the query q_s. Here, we suppose that $rel(q_s, t_i) > rel(q_s, t_j)$, indicating that t_i is more relevant to the query q_s than t_j, that is, t_i could better describe the relationship r_k between the entities e_h and e_t mentioned in q_s. Formally, given a triplet (q_s, t_i, t_j), our goal is to learn a representation function $v(\cdot)$ for q_s, t_i, and t_j, and use the learned vector representations to calculate the similarity between each title and the query, such that given q_s, the more relevant title t_i can achieve higher similarity score:

$$\begin{aligned} &S(v(q_s), v(t_i)) > S(v(q_s), v(t_j)), \\ &\forall q_s, t_i, t_j \text{ such that } rel(q_s, t_i) > rel(q_s, t_j), \end{aligned} \tag{1}$$

where the similarity $S(\cdot, \cdot)$ is calculated by the cosine similarity between two vectors, and it is also used in [Huang *et al.*, 2013] to compute the relevance score between a document and a query as cosine similarity of their corresponding semantic concept vectors.

As the relevance scores exhibit a relative ranking order for two pairs, a ranking layer on the top is employed to evaluate the loss of a triplet (q_s, t_i, t_j). Here, the margin ranking loss [Herbrich *et al.*, 1999] is used, which is a convex approximation to the 0-1 ranking error loss, and it is defined as follows:

$$Loss(q_s, t_i, t_j) = max(0, 1 - S(v(q_s), v(t_i)) + S(v(q_s), v(t_j))). \tag{2}$$

[4] To comply with the company's non-disclosure policy, we normalized all the values of clicks. The examples are translated from Chinese.

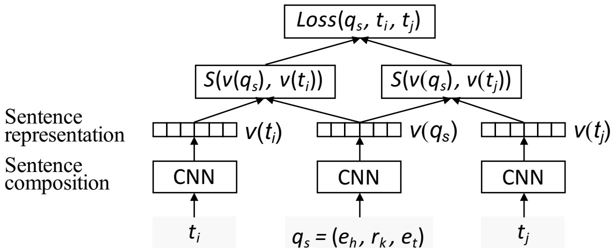


Figure 2: The architecture of pairwise ranking model.

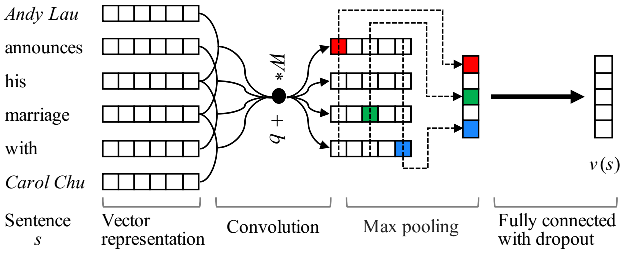


Figure 3: The CNN architecture for sentence composition.

The ranking layer does not have any parameters. During learning, it evaluates the model's violation of the ranking order specified in the triplet, and back-propagates the gradients to the lower layers so that these layers can adjust their parameters to minimize the ranking loss.

Sentence Representation

The architecture of the CNN used for sentence composition is shown in Figure 3. Given a sentence s, a CNN is employed to obtain a distributed representation $v(s)$ of s. First, words in the sentence are transformed into vector representations via word embedding matrix, which capture semantic information of words. Second, in the convolutional layer, a set of filters is applied to a sliding window of length h (3 is used for demonstrating in Figure 3) over the sentence to extract a set of local features. To ensure that the filters can be applied to every element of the input matrix, zero-padding is applied to the input at both ends prior to convolution. The filters are automatically learned during the training phase of the neural network. Third, a max pooling layer performs max operation over a local neighborhood to retain only the most useful local features produced by the convolutional layer. Finally, the output of the max pooling layer is passed to a fully connected layer, which computes a non-linear transformation of these local features, here the sigmoid activation function is used.

Training

The network is trained by minimizing an objective function over the training set. More specifically, given a set of triplets $\mathcal{P} = \{(q_s, t_i, t_j)\}$, the representation function $v(\cdot)$ can be learned by minimizing the following objective function:

$$\min_{W} \sum_{(q_s,t_i,t_j)\in\mathcal{P}} Loss(q_s, t_i, t_j) + \lambda||W||^2, \qquad (3)$$

where λ is a regularization parameter used to improve the generalization of the learned ranking model. W is the parameters of the representation function $v(\cdot)$. To prevent neural networks from overfitting, dropout [Srivastava *et al.*, 2014] with the probability of 0.5 is applied to all the fully connected layers of the network, and L2 regularization is applied over the neural network weights.

Sentence Ranking

The learned representation function $v(\cdot)$ maps sentences to feature vectors in a latent semantic space, we can use it to generate high-level semantic representations for a test query q_t and all of its candidate sentences $S_t = \{s_1, s_2, \cdots, s_n\}$. Then, the sentences in S_t are ranked by comparing the similarity score between each sentence s_i and q_t. Here, the test query q_t is also formed by concatenating two test entities e_i, e_j and their relationship r_k with the order $e_i\ r_k\ e_j$.

4 Experiments

This section describes the dataset, evaluation metrics, baselines, and parameter settings in our experiments.

4.1 Dataset and Evaluation Metrics

Our method is not restricted in domain or language, since the ranking model employed here is language independent, and the features are learned automatically. In this paper, the proposed model and all baseline models are trained on a Chinese corpus. Following the previous work [Voskarides *et al.*, 2015], we also take "people" entities as a case study and evaluate all models on this type of entities and relationships between them. Using the training data acquisition method described in Section 3.1, we generate 1,596,489 training examples (denoted as $\mathcal{P}_r$) for the pairwise models. Following previous work [Huang *et al.*, 2013], we collect the training data for the pointwise models by extracting clicked/unclicked titles as positive/negative examples, respectively. In this way, we collect 71,536 positive and 131,334 negative examples (denoted as $\mathcal{P}_t$) from the identical clickthrough data.

We constructed the test set as follows. First, we randomly sampled a set of 20,000 relation triplets consisting of "people" entities and their relationships from a proprietary knowledge graph that is used by Baidu Web search engine. To conduct fair evaluation, we filtered the triplets that appear in our training data. A total of 9,702 triplets were left, denoted as $R = \{(e_i, r_k, e_j)\}$, in which e_i and e_j are a pair of entities, r_k is the relationship between them. To comply with the baselines, we followed the method in [Voskarides *et al.*, 2015] that retrieves candidate sentences for each triplet in R from the largest Chinese online encyclopedia Baidu Baike[5]. In which, it is required that the candidate sentences must contain both e_i and e_j, as well as r_k or a related term of r_k. Here related terms of r_k are synonyms of r_k, or similar phrases of r_k measured by the cosine similarity between their word embeddings [Mikolov *et al.*, 2013]. This results in a total of 8,461 sentences for the triplets in R.

Then, we asked three annotators to manually label the quality of sentences for each triplet in R as per a five-level graded relevance scale: *perfect*, *excellent*, *good*, *fair*, and *bad*. Similar to [Voskarides *et al.*, 2015], the judging was based on how

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

well a sentence describes the relationship for an entity pair. To examine the agreement between the annotators, we compute the kappa [Carletta, 1996] statistic between them. The kappa statistic is 0.65, which indicates a substantial agreement (K: 0.61-0.8) according to [Landis and Koch, 1977]. In our experiments, we kept the sentences with at least two agreements out of three for use, and a total of 8,046 sentences were obtained.

Finally, as is common in information retrieval evaluation, we discarded the triplets in R if the number of sentences w.r.t. a triplet is less than 2. In addition, we filtered the triplets whose candidate sentences are all labeled the same grade, since they are useless for evaluating the ranking models. This results in a total of 4,771 candidate sentences for 921 relation triplets, which are denoted as T and used as the test set. Statistics show that T covers 69 unique relation types. The numbers of candidate sentences for each triplet range from 2 to 45, with an average of 5.2 per triplet. Of all relevance grades, 17.77% is *perfect*, 6.08% is *excellent*, 24.36% is *good*, 11.38% is *fair*, and 40.41% is *bad*.

Following previous work [Voskarides *et al.*, 2015], we also evaluate the methods according to three metrics: NDCG [Jarvelin and Kekalainen, 2002], ERR [Chapelle *et al.*, 2009], Exc@1 and Per@1. NDCG and ERR are commonly used metrics for evaluating the ranking models in information retrieval. Exc@1 and Per@1 are used to evaluate whether the methods could rank the best possible sentence (a *perfect* or *excellent* sentence) at the top of the ranking results.

4.2 Baseline Methods

We evaluate the proposed method against several strong baselines. Specially, the following three methods are selected for comparison. First, we compare our method with [Voskarides *et al.*, 2015], which used a pointwise learning to rank approach with handcrafted features and employed a Random Forest classifier [Breiman, 2001], which is referred to as B1-RF hereafter. The second baseline is a pointwise ranking model with CNNs for ranking short text pairs proposed by [Severyn and Moschitti, 2015], which is referred to as B2-TR. The third baseline uses GBRank [Zheng *et al.*, 2007] to train a pairwise ranking model with handcrafted features (denoted as B3-GB). We use the same pairwise ranking loss defined in Equation 2 for GBRank.

4.3 Parameter Settings

Given the pairwise training data $\mathcal{P}_r$ acquired in Section 4.1, it was randomly split into training set $\mathcal{P}_r^t$ (80%) and validation set $\mathcal{P}_r^v$ (20%). $\mathcal{P}_r^t$ was used to train B3-GB and the proposed model, and $\mathcal{P}_r^v$ was used to tune the parameters for both models. On the other hand, $\mathcal{P}_t$ was split into $\mathcal{P}_t^t$ (80%) and $\mathcal{P}_t^v$ (20%) for the training and parameter tuning of the two pointwise baselines B1-RF and B2-TR. We used a grid search to determine the optimal parameters of each model in the feasible space of selected parameters. For B1-RF, we selected the number of trees among $\{100, 200, \cdots, 1000\}$, sub-sampling rate and feature sampling rate both among $\{0.1, 0.2, \cdots, 1.0\}$. For B3-GB, we selected the number of trees among $\{200, 250, \cdots, 1000\}$, learning rate among $\{0.0001, 0.001, 0.01, 0.1\}$, and tree depth among $\{3, 4, 5\}$. For B2-TR and our model, we selected learning rate for SGD among $\{0.0001, 0.001, 0.01, 0.1\}$, the dimension of word embeddings and the sentence embedding size both among $\{50, 100, \cdots, 300\}$, the sliding window size among $\{1, 3, 5, 7\}$, and the batch size among $\{64, 128, \cdots, 1024\}$. To tune each model, the NDCG score on corresponding validation set was used to evaluate the performance of a trained model with the given parameters. Finally, the models with the parameter settings that best performed on the validation set were selected.

Parameter	PR-CNN	B2-TR
Batch size	256	256
Learning rate	0.01	0.01
Sliding window size	3	3
Sentence embedding size	200	200
Dimension of word embeddings	200	150

Table 2: Parameter settings for PR-CNN and B2-TR.

Method	NDCG@1	NDCG@10	ERR@1	ERR@10
B1-RF	0.4549	0.7496	0.2492	0.4198
B2-TR	0.5466	0.7999	0.3593	0.4995
B3-GB	0.5639	0.8119	0.3448	0.4929
PR-CNN	**0.6285**	**0.8370**	**0.3747**	**0.5121**

Table 3: Performance of the baselines and our method.

The parameters used in the experiments are as follows. For B1-RF, the number of trees, sub-sampling rate, and feature sampling rate are 100, 0.9, and 0.1, respectively. For B3-GB, the number of trees, learning rate, and tree depth are 1000, 0.1, and 5, respectively. For B2-TR and our model, the parameters used in the experiments are listed in Table 2.

5 Results and Analysis

We used the trained three baseline models and our model to rank sentences for each triplet in the test set T as described in Section 4.1. In what follows, we compare the performance of these methods, and address the following issues. First, we evaluate the performance of different methods in ranking sentences for explaining entity relationships with or without handcrafted features. Second, we compare the ranking models based on pairwise or pointwise learning. Third, we investigate whether our method could be applied to rank sentences beyond Baike, such as sentences retrieved from Web pages.

5.1 Evaluation of the Methods

Table 3 shows the evaluation results of each method. Boldface indicates the highest score w.r.t. each metric. From the results in Table 3, we have some interesting findings.

First, we can see that PR-CNN achieves the highest NDCG and ERR scores in comparison with the three strong baselines, which shows that PR-CNN is more effective in ranking sentences to describe the relationship between two entities. Especially, PR-CNN significantly outperforms the baseline B1-RF by a large margin in terms of both NDCG and ERR,

Has one	# triplets	# sentences	Method	NDCG@1	NDCG@10	ERR@1	ERR@10	Exc@1	Per@1
perfect	549	3,098	B3-GB	0.6084	0.8352	0.4943	0.6825	0.5118	0.4645
			PR-CNN	**0.6594**	**0.8538**	**0.5289**	**0.7045**	**0.5501**	**0.4882**
excellent	251	1,267	B3-GB	0.5920	0.8321	0.3357	0.5019	0.4741	–
			PR-CNN	**0.6677**	**0.8616**	**0.3787**	**0.5322**	**0.5777**	–
good	607	3,544	B3-GB	0.5699	0.8083	0.3189	0.4633	–	–
			PR-CNN	**0.6135**	**0.8283**	**0.3331**	**0.4773**	–	–
fair	266	1,859	B3-GB	0.5182	0.7728	0.2707	0.4150	–	–
			PR-CNN	**0.5667**	**0.8007**	**0.2953**	**0.4384**	–	–

Table 4: The evaluation results of different relevance grades for the best baseline (B3-GB) and our method (PR-CNN).

Method	Perfect	Excellent	Good	Fair	Bad
B1-RF	13.03%	11.40%	42.13%	6.19%	27.25%
B2-TR	15.64%	9.23%	39.30%	6.19%	29.64%
B3-GB	22.15%	9.56%	36.70%	5.21%	26.38%
PR-CNN	**24.43%**	**12.81%**	36.27%	7.71%	18.78%

Table 5: The evaluation results on sentences from Web pages.

which demonstrates that our ranking method can achieve substantially higher performance in ranking sentences than the original method proposed for this task.

Second, we compare ranking models with or without handcrafted features. The results reveal that both in terms of NDCG and ERR, B2-TR significantly outperforms B1-RF, and PR-CNN significantly outperforms B3-GB. B1-RF and B3-GB both design a rich set of handcrafted features, while B2-TR and PR-CNN both employ CNNs to automatically learn relevant features. This demonstrates that ranking models with neural networks perform significantly better than handcrafted features based ranking models. The performance of the models with handcrafted features depends strongly on the quality of manually designed features. Since errors exist inevitably in the feature extraction process, the use of handcrafted features leads to error propagation. By contrast, ranking models with neural networks can automatically learn features directly from the training examples by using CNNs. Automatically learning features via CNNs can alleviate the error propagation that occurs in traditional feature extraction [Zeng *et al.*, 2015].

Third, we compare different ranking models based on pairwise or pointwise learning. As shown in Table 3, PR-CNN significantly outperforms B2-TR, and B3-GB significantly outperforms B1-RF. PR-CNN and B3-GB are both pairwise ranking approaches trained on $\mathcal{P}_r$, while B2-TR and B1-RF are both pointwise ranking approaches trained on $\mathcal{P}_t$. This suggests that pairwise ranking approaches are better suited for this task than pointwise approaches. The main reason is that, the accuracy of absolute relevance judgments would inevitably be affected by the noise contained in clickthrough data [Joachims *et al.*, 2007]. Compared with absolute relevance judgments, relative relevance judgments extracted from clickthrough data contain many subtle differences between titles which are useful for learning an accurate and stable ranking. Similar findings were also reported by [Joachims *et al.*, 2005], which showed that click based relative preference derived from clickthrough data is more accurate than absolute preference. Therefore, the relative order among sentences can be better modeled by exploiting the relative relevance preferences in training data with pairwise approaches [Liu, 2009].

Finally, compared with all baselines, PR-CNN can best achieve the goal of ranking sentences for explaining relationships between entities. Specially, the results in Table 3 demonstrate that PR-CNN significantly outperforms all other methods by a large margin in terms of NDCG@1 and ERR@1, which indicates that PR-CNN is more effective in ranking the best possible sentence at the top of the ranking results. To get insight into the performance of different triplets, all triplets in the test set T are further grouped into different grades by if a particular relevance grade is held by at least one sentence of a triplet. Table 4 shows the evaluation results of different groups for PR-CNN and the best baseline B3-GB. Results show that PR-CNN significantly outperforms B3-GB by a large margin in terms of *perfect*, *excellent*, Exc@1, and Per@1, which verifies that PR-CNN outperforms the best baseline in ranking sentences for triplets that have at least one high-quality candidate sentence. In conclusion, the evaluations demonstrate that PR-CNN can provide sentences of highest quality to end users with the restriction that only one sentence can be shown for an entity pair.

5.2 Evaluation on Web Pages

In this section, we investigate whether the model trained with data from search logs can be applied to extract entity relationship descriptions from Web pages. First, from more than 6 millions of Web pages, we extracted up to 1,000 candidate sentences for each triplet in T using the method described in Section 4.1. Then, we used the trained three baseline models and our model to rank sentences for each triplet. Finally, we evaluated the top-ranked 1 sentence for each triplet using the method described in Section 4.1, and the percentage of each grade was calculated. Table 5 shows the evaluation results. We can see from the table that PR-CNN achieves the highest scores in terms of *Perfect* and *Excellent* in comparison with the three baselines, which shows that PR-CNN can also perform best in ranking sentences retrieved from Web pages.

5.3 Case Study

To further analyze our model, we conduct a case study. We show several representative examples of the proposed model, in comparison with the baseline models. Figure 4 gives the top-ranked 1 sentences for each triplet of different models,

Example a

Triplet: (吴奇隆, 前妻, 马雅舒)

(Nicky Wu, Former spouse, Ma Yashu)

Sentences

B1-RF/B2-TR: 2001年吴奇隆和马雅舒通过拍摄《萧十一郎》相识。

Nicky Wu and Ma Yashu met each other when starring in Xiao Shiyi Lang in 2001.

B3-GB: 同年8月11日, 吴奇隆与马雅舒正式办理离婚手续。

Nicky Wu and Ma Yashu got a divorce on August 11 in the same year.

PR-CNN: 2009年8月11日马雅舒与吴奇隆正式办理离婚手续。

Ma Yashu and Nicky Wu got a divorce on August 11, 2009.

Example b

Triplet: (李敖, 女儿, 李文)

(Li Ao, Daughter, Li Wen)

Sentences

B1-RF/B2-TR/B3-GB: 李敖得知很高兴, 给她取名李文。

Li Ao was very glad to hear that and named her Li Wen.

PR-CNN: 李文是李敖与当年台大校花王尚勤的女儿, 是李敖的长女。

Li Wen is the daughter of Li Ao and Wang Shangqin who once was a school beauty in NTU, and she is the eldest daughter of Li Ao.

Figure 4: Examples of the top-ranked 1 sentences by different models. We provide literal English translations for the triplets and the sentences.

and it shows that PR-CNN succeeds in ranking a *perfect* sentence at the top of the ranking results. As can be seen from the examples, our model PR-CNN learns from the training data the phrases that are likely to provide better evidence for explaining a certain relationship. For example, the phrase "got a divorce" is a better clue to explain the relationship "Former spouse" between two people than the phrase "met each other" as shown in example a. Compared with the phrase "named her" in example b, "the eldest daughter" provides more detailed evidence about the relationship "Daughter" between the two entities.

We also conduct error analysis. Two examples are given in Figure 5. From both examples, we can find that PR-CNN fails to provide a high-quality sentence (a *perfect* or *excellent* sentence) at the top of the ranking results for the given triplet. The reasons for this are two-fold. First, 25.84% triplets in the test set have no high-quality candidate sentences, example c shows such a case. This suggests that it is necessary to improve the performance of candidate sentences retrieval for the triplets, which we leave in future work. Second, 46.12% triplets in the test set that have at least one high-quality candidate sentence fail to get a best possible sentence ranked at the top of the ranking results, example d shows such a case. A potential solution to this issue is to use some kind of unsupervised method to refine the candidate sentences. For example, we can cluster all the candidate sentences describing a given relation type, and then filter those that are semantically far from all the bigger clusters.

Example c

Triplet: (曹颖, 丈夫, 王斑)

(Cao Ying, Husband, Wang Ban)

Sentences

PR-CNN: 在王斑心中, 曹颖是一个电视上和生活中区别不大的人。

In Wang Ban's mind, Cao Ying's personality in real life is almost the same as on TV.

The *perfect* or *excellent* sentence: N/A

Example d

Triplet: (刘德华, 前女友, 喻可欣)

(Andy Lau, Ex-girlfriend, Yu Kexin)

Sentences

PR-CNN: 喻可欣母亲上节目大骂刘德华 爆华仔欺骗朱丽倩与前女友。

On a TV show, Yu Kexin's mother accused Andy Lau of lying to both Carol Chu and his ex-girlfriend.

The *perfect* sentence: 喻可欣曾说和刘德华论及婚嫁, 是因为朱丽倩当小三, 刘德华才和她分手。

Yu Kexin said that she and Andy Lau would have got married if it were not for Carol Chu.

Figure 5: Error analysis for the PR-CNN model.

6 Conclusions and Future Work

In this paper, we study the problem of explaining relationships between entities, and evaluate its effectiveness on manually annotated sentences extracted from both Baike documents and Web pages. We model the task as a ranking problem, and propose an effective pairwise ranking model with neural networks to rank the sentences based on automatically learned features. The experiments show that our method significantly outperforms several strong baselines.

As future work, we plan to evaluate how our method performs on entities and relationships of any type and popularity. We are also interested in exploring ways to explain changes in relationships over time. Furthermore, if we want to apply the obtained results to support applications for a search engine, the quality and readability of sentences need to be further improved.

Acknowledgments

This research is supported by the National Basic Research Program of China (973 program No. 2014CB340505). We would like to thank the anonymous reviewers for their insightful comments.

References

[Agichtein *et al.*, 2006] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR*, 2006.

[Banko *et al.*, 2007] M. Banko, M. J Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

[Blanco and Zaragoza, 2010] R. Blanco and H. Zaragoza. Finding support sentences for entities. In *SIGIR*, 2010.

[Blanco *et al.*, 2013] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *ISWC*, 2013.

[Bordes *et al.*, 2013] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2013.

[Breiman, 2001] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Carletta, 1996] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 1996.

[Chapelle *et al.*, 2009] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM*, 2009.

[Dou *et al.*, 2008] Z. Dou, R. Song, X. Yuan, and J.-R. Wen. Are click-through data adequate for learning web search rankings? In *CIKM*, 2008.

[Fader *et al.*, 2011] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP*, 2011.

[Fang *et al.*, 2011] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. REX: explaining relationships between entity pairs. *Proceedings of the VLDB Endowment*, 5(3), 2011.

[Gao *et al.*, 2010] J. Gao, X. He, and J.-Y. Nie. Clickthrough-based translation models for web search: from word models to phrase models. In *CIKM*, 2010.

[Herbrich *et al.*, 1999] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Neural Information Processing Systems*, 1999.

[Huang *et al.*, 2013] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

[Huang *et al.*, 2016] J. Huang, S. Zhao, S. Ding, H. Wu, M. Sun, and H. Wang. Generating recommendation evidence using translation model. In *IJCAI*, 2016.

[Jarvelin and Kekalainen, 2002] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[Ji *et al.*, 2016] G. Ji, K. Liu, S. He, and J. Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*, 2016.

[Joachims *et al.*, 2005] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, 2005.

[Joachims *et al.*, 2007] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), 2007.

[Joachims, 2002] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.

[Kim, 2014] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[Landis and Koch, 1977] J R. Landis and G. G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), March 1977.

[Lin *et al.*, 2015] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[Liu, 2009] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[Mikolov *et al.*, 2013] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 2013.

[Santos *et al.*, 2015] C. N. D. Santos, B. Xiang, and B. Zhou. Classifying relations by ranking with convolutional neural networks. In *ACL-IJCNLP*, 2015.

[Severyn and Moschitti, 2015] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, 2015.

[Singhal, 2012] A. Singhal. Introducing the knowledge graph: things, not strings. Official Blog of Google: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html, 2012.

[Socher *et al.*, 2012] R. Socher, B. Huval, C. Manning, and A. Ng. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, 2012.

[Srivastava *et al.*, 2014] N. Srivastava, G. E Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Voskarides *et al.*, 2015] N. Voskarides, E. Meij, M. Tsagkias, M. De Rijke, and W. Weerkamp. Learning to explain entity relationships in knowledge graphs. In *ACL-IJCNLP*, 2015.

[Wang *et al.*, 2014] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[Yu *et al.*, 2014] X. Yu, H. Ma, B. P. Hsu, and J. Han. On building entity recommender systems using user click log and freebase knowledge. In *WSDM*, 2014.

[Zeng *et al.*, 2015] D. Zeng, K. Liu, Y. Chen, and J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, 2015.

[Zhao *et al.*, 2010] S. Zhao, H. Wang, and T. Liu. Paraphrasing with search engine query logs. In *COLING*, 2010.

[Zheng *et al.*, 2007] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR*, 2007.