Aggregating Inter-Sentence Information to Enhance Relation Extraction

Hao Zheng[⋆], **Zhoujun Li**[⋆], **Senzhang Wang**[⋆], **Zhao Yan**[⋆], **Jianshe Zhou**[†]
[⋆] State Key Laboratory of Software Development Environment,
Beihang University, Beijing, 100191, China
[†] Beijing Advanced Innovation Center for Imaging Technology,
Capital Normal University, 100048, China
{zhenghao35, lizj, szwang, yanzhao}@buaa.edu.cn, zhoujs@cnu.edu.cn

Abstract

Previous work for relation extraction from free text is mainly based on *intra-sentence* information. As relations might be mentioned across sentences, *inter-sentence* information can be leveraged to improve distantly supervised relation extraction. To effectively exploit inter-sentence information , we propose a ranking-based approach, which first learns a scoring function based on a listwise learning-to-rank model and then uses it for multi-label relation extraction. Experimental results verify the effectiveness of our method for aggregating information across sentences. Additionally, to further improve the ranking of high-quality extractions, we propose an effective method to rank relations from different entity pairs. This method can be easily integrated into our overall relation extraction framework, and boosts the precision significantly.

Introduction

Relation extraction (RE) aims to generate structured relational knowledge from unstructured natural language text. Traditional supervised approaches for relation extraction (RE), based on hand-labeled corpora, cannot satisfy the demand for web-scale relation extraction due to the expensive and laborious annotation. To address this issue, Mintz et al. (2009) proposed a *distant supervision (DS)* based approach to automatically generate labeled data by heuristically aligning a database of relational facts (e.g., Freebase) with free text (e.g., New York Times corpora). An example accounting for this process is shown in Table 1. In this paper, we define that a instance is a sentence mentioning a certain entity pair in knowledge bases. There might exist multiple matched instances for the same entity pair and this case is called "multi-instance". In Table 1, we label the matched instances for the entity pair *(Steven Spielberg, Saving Private Ryan)* using the gold standard relation coming from knowledge bases *Film-director (Steven Spielberg, Saving Private Ryan)*.

However, distant supervision still faces the following problems. First, a sentence that mentions two entities participating in a relation in knowledge bases would not necessarily express this relation explicitly. Also, there may be multiple relations holding for the same pair of entities (this is called "multi-label") and it's hard to make out which relation a sentence that mentions the entity pair should be labeled by.

Entity pair	(Steven Spielberg, Saving Private Ryan)
Gold relation from knowledge bases	**Film-director (Steven Spielberg, Saving Private Ryan)**
Relation instances from free text	**S1: Steven Spielberg**'s film **Saving Private Ryan** is loosely based on the brothers story. **S2:** Allison co-produced the Academy Award-winning **Saving Private Ryan**, directed by **Steven Spielberg** ...

Table 1: Labeled instances by distant supervision, using the relation *Film-director (Steven Spielberg, Saving Private Ryan)* in knowledge bases

In their efforts to address these limits, researchers have followed two fundamentally different lines, which we call *entity-level learning* and *sentence-level learning* , respectively. The original work (Mintz et al. 2009; Riedel, Yao, and McCallum 2010) in DS for RE fell into the entity-level learning, which aggregated features from multiple sentences to directly make entity-level extraction (not assign relation labels to individual sentences). However their approaches ignored detailed sentence-level information, and could not deal with the multi-label problem. The subsequent work (Hoffmann et al. 2011; Surdeanu et al. 2012; Ritter et al. 2013) used the latent-variable to explicitly model the sentence-level relation which is assigned to a single sentence. These approaches all assign certain labels to individual sentences, which intrinsically fell into the sentence-level learning. While they have improved precision and recall in relation extraction, they are all based on the inaccurate built-in assumption that a sentence expresses either one certain relation or no relation. As relations also can be mentioned across sentences, this assumption may not always hold, which partly causes the loss of some indicative information about the types of relation. For example, Mintz et al. (2009) showed that combining information in different matched sentences for the same entity pair can help predict its relations.

From Table 1, we can see that relying on the sentence S1 alone cannot certainly decide the *Film-director* relation,

since it could instead be the evidence for *Film-writer* or *Film-producer* relation. The sentence S2 also could be evidence for other relations which are expressed in a similar way (e.g., Apple, directed by Jobs . . .). However, the combination of the two sentences is a strong indicator of the *Film-director* relation. Therefore, conducting relation prediction by aggregating inter-sentence information could capture more information otherwise a single sentence may not contain, and potentially improve the performance. In this paper, we argue that entity-level learning is a more promising and appropriate line compared with sentence-level learning due to the existence of natural groups (groups of instances for the same entity pair) in DS scenario and the significance of aggregating inter-sentence information.

We propose an effective entity-level relation extraction learning algorithm called RankRE to aggregate the inter-sentence information. Based on a novel perspective of learning to rank, RankRE can deal with the special multi-instance and multi-label learning problem (Surdeanu et al. 2012) in the DS scenario. More importantly, it effectively leverages inter-sentence information to further enhance relation extraction. The intuition of our method is that, it is more natural for a sentence to rank all possible relations based on the likelihood that this sentence explicitly or implicitly expresses them, than just assign a single label to it. For example, given the sentence S1 in Table 1, relations *Film-writer*, *Film-producer* and *Film-director* should be ranked at higher positions than other irrelevant relations based on the likelihood that the sentence S1 explicitly or implicitly expresses the corresponding relation. We do not make an aggressive relation prediction for any instance (as it might be wrong), but rather learn a scoring function that maps each relation class to a real value which represents the likelihood. Finally, we make predictions of relation labels for a pair of entities by aggregating ranking scores from all its matched instances. Additionally, as the high-quality extractions are particularly important in web-scale relation extraction, we propose a method from a "global" ranking perspective to further improve the ranking of high-quality extractions. Specifically, this method imposes ranking constrains on relations from different entity pairs and successfully ranks the most confident extractions at the highest positions.

We conduct experiments on a widely used dataset from two different aspects. Experimental results demonstrate that our approach outperforms other baselines in both relation extraction precision and scalability.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, RankRE is the first RE approach that models the multi-instance and multi-label problem in the DS scenario by learning to rank. We exploit this technique to aggregate inter-sentence information and conduct entity-level learning.
- RankRE is computationally efficient, and runs much faster than existing methods in distantly supervised relation extraction.
- We verify the efficiency and effectiveness of our model through two different levels of experiments.

The remainder of this paper is organized as follows: we start by discussing related work. Then we describe our ranking-based approach for distantly supervised relation extraction and analyze how this approach can capture inter-sentence information. Finally, we evaluate our approach and report the results.

Related Work

Distant Supervision for Relation Extraction

The idea of distant supervision was first introduced in the field of bioinformatics (Craven, Kumlien, and others 1999). Mintz et al. (2009) adopted distant supervision to perform large-scale relation extraction by using Freebase (a knowledge base) to distantly or weakly annotate Wikipedia corpora. Riedel et al. (2010) proposed the At-Least-One assumption which relaxed the strong DS assumption. As described in our introduction part, these methods can exploit aggregate information from multiple sentences, whereas it ignores sentence-level detail and cannot conduct multi-label extraction. We aim at overcoming these disadvantages and exploiting aggregate information more effectively. To resolve the multi-label problem, methods which annotated every instance with a latent variable were proposed (Hoffmann et al. 2011; Surdeanu et al. 2012; Ritter et al. 2013). Han et al. (2014) proposed a novel semantic consistency model, which achieved impressive performance in extracting relations of long tail instances.

Another active research direction for distant supervision is to directly reduce the noise in training data. Takamatsu et al. (2012) proposed a generative model to eliminate false positive training examples. Xu et al. (2013) proposed a passage retrieval model to reduce false negative training examples. Roller et al. (2015) detected potential false negative training examples by using a knowledge inference method.

Our work is closely related to Weston et al. (2013) which also adopted a ranking method for relation extraction. Their approach used a ranking-based embedding framework which was inspired by TransE (Bordes et al. 2013), a promising embedding method modeling multi-relational data. However, they still learned the scoring function based on single instance, and thus, like other approaches described above, cannot exploit aggregate information from multiple instances during training. Besides, their method cannot effectively handle the multi-label problem.

Learning to Rank

Learning to rank, a recently popular machining learning technique, is widely used in document retrieval, collaborative filtering and many other applications whose central problem is ranking (Herbrich, Graepel, and Obermayer 1999; Joachims 2002; Freund et al. 2003; Burges et al. 2005). In this section, we briefly review the work concerned with our work.

Our method is inspired by the listwise approach in learning to rank (Cao et al. 2007; Xia et al. 2008). Their approaches learned a ranking function by taking individual lists, rather than pairs, as instances and minimizing a loss

function defined over them. We adapt their listwise approaches for this problem by slightly changing the way of transforming scores to probability distributions. Also, we adopt a different learning method, a perceptron-style additive parameter update approach (Collins 2002), which turns out to be efficient and effective.

Our Method

In this section, we describe our ranking-based method for relation extraction. We first model the likelihood that a instance explicitly or implicitly expresses each relation of interest by learning to rank; then we learn the ranking function and exploit it for relation extraction.

Relation Likelihood Modeling via Learning to Rank

We assume that there exists a set of relation types of interest R. Given an entity pair (e_1, e_2), we denote by $S = \{r_1, ..., r_m\}(S \subset R)$ the set of gold standard relations for this entity pair in the knowledge base, and denote by $X = \{x_1, ..., x_n\}$ the set of matched instances for this pair in the free text. We define a scoring function f, which outputs a score $f(x_j, r_i)$ measuring the likelihood that the instance x_j expresses or implicates the relation $r_i (r_i \in R)$. Then all evidence for a candidate relation r_i is combined as follows:

$$f_i = \sum_{j=1}^{n} f(x_j, r_i) \tag{1}$$

where f_i represents the accumulated score of relation r_i from all matched instances. Then we assume that the m relations in the set S should be ranked in the top m in all candidate relations according to the accumulated score f_i:

$$\forall r_i \in S, \forall r_{i'} \in R - S : f_i > f_{i'} \tag{2}$$

Motivated by the learning-to-rank model (Cao et al. 2007; Xia et al. 2008), we define the "top m" probability distributions based on aforementioned scores as follows:

$$P_m(r_{l_1}, ..., r_{l_m}) = \frac{e^{\sum_{i=1}^{m} f_{l_i}}}{\sum_{r_{s_1}, r_{s_2}, ..., r_{s_m} \in R} e^{\sum_{i=1}^{m} f_{s_i}}} \tag{3}$$

which represents the probability that m relations $r_{l_1}, r_{l_2}, ..., r_{l_m}$ are placed at the top m positions. In practice, each instance-relation pair (x_j, r_i) is represented as a feature vector and the scoring function $f(x_j, r_i)$ is defined as a simple linear function as follows:

$$f(x_j, r_i) = \mathbf{w}^T \mathbf{\Phi}(x_j, r_i) \tag{4}$$

where $\mathbf{w}$ is the parameter vector of function f and $\mathbf{\Phi}(x_j, r_i)$ is the feature vector representation of the instance-relation pair (x_j, r_i). More complicated scoring function (say a neural network) could also be used to further boost the performance, yet in this paper we just choose the simple linear function. The goal of the learning process is to do optimization such that the m gold standard relations in the set $S(S = \{r_1, ..., r_m\})$ are ranked at the top m positions. To do that, the probability $P_m(r_1, r_2, ..., r_m)$ is maximized to estimate the parameter vector $\mathbf{w}$ during training.

The main advantage of our model is that it effectively aggregates valuable information coming from each instance for final relation prediction, even although the instance does not directly express a certain relation. For example, both the sentence S1 and S2 in Table 1 rank the relation *Film-director* as one of the top positions, whereas only combining the two sentences would rank the relation *Film-director* at the first position.

Another advantage of our model is the robustness to the false positive training examples, which could be equally important in the DS scenario due to the large amounts of mislabeled data. Some noisy instances among all instances for an entity pair are allowed to obtain low scores for all relations, as long as the overall scores for all relations have a desired ranking. Therefore, our approach is robust to noise by relaxing the strong DS assumption.

Algorithm 1 The learning algorithm for RankRE-local

Input:
The training dataset $D = \{(X^q, S^q)|q = 1, \ldots, M\}$ consisting of (instance set, relation set) pairs.
A parameter T specifying the number of iterations over the training set.

Output:
The parameter vector $\mathbf{w}$.

For convenience, we define the summed feature vector $\mathbf{\Psi}_r^q = \sum_{x \in X^q} \mathbf{\Phi}(x, r)$
initialize parameter vector $\mathbf{w} \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  for $q = 1, \ldots, M$ **do**
    $m \leftarrow |S^q|$
    $S_\star^q \leftarrow \text{top\_}m(X^q)$ [1]
    for each $r \in S^q$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{\Psi}_r^q$
    end for
    for each $r \in S_\star^q$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{\Psi}_r^q$
    end for
  end for
end for
return $\mathbf{w}$

Training Method and Implementation Details

Training is fairly challenging, for the probability object is difficult to optimize. The algorithm would be very time consuming to find an exact solution, since the number of terms in the denominator grows exponentially with the number of the gold standard labels. We instead use a perceptron-style additive parameter update schema to indirectly optimize the object (Collins 2002). This parameter estimation algorithm is computationally efficient, and also works well in practice. Specifically, the training algorithm takes T passes over the training dataset. All parameters are initially set to be zeros. For each entity pair, the top m scoring relations are predicted under the current parameter setting. If these m relations do not match with the m gold standard relations for this entity pair, the parameter vector is updated in a simple additive

Algorithm 2 The learning algorithm for RankRE-global

Input: $D = \{(X^q, S^q)|q = 1, \ldots, M\}, T$
Output: $\mathbf{w}$

initialize parameter vector $\mathbf{w} \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  for $q = 1, \ldots, M$ **do**
    $m \leftarrow |S^q|$
    $S^q_\star \leftarrow \text{top\_}m(X^q)$
    for each $r \in S^q$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{\Psi}^q_r$
    end for
    for each $r \in S^q_\star$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{\Psi}^q_r$
    end for
    select at random a training pair $(X^{q'}, S^{q'})(|X^{q'}| = |X^q|)$
    for each $r \in S^q \wedge r \notin S^{q'}$ **do**
      if $\mathbf{w}^T\mathbf{\Psi}^q_r < \mathbf{w}^T\mathbf{\Psi}^{q'}_r$ **then**
        $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{\Psi}^q_r$
        $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{\Psi}^{q'}_r$
      end if
    end for
  end for
end for
return $\mathbf{w}$

fashion. The algorithm is detailed in Algorithm 1.

In practice, we adopt the "averaged parameters" method, which, as reported by Collins (2002), performs significantly better than the standard one. Formally, the final returned parameter vector is defined as:

$$\mathbf{w} = \frac{1}{TM}\sum_{t=1}^{T}\sum_{q=1}^{M}\mathbf{w}^{t,q} \tag{5}$$

where $\mathbf{w}^{t,q}$ is the parameter vector after the qth training example has been processed in pass t over the training dataset. The original algorithm in Algorithm 1 can be easily modified to compute the averaged parameter vector.

Implementation for Multi-label Relation Extraction

This section describes how to exploit the scoring function f for multi-label relation extraction.

Given a set of matched instances $X = \{x_1, ..., x_n\}$ for an entity pair (e_1, e_2), we define the overall likelihood of assigning a relation r_i to this entity pair as:

$$lik(X, i) = \frac{a \cdot \log(n) + 1}{n}\sum_{j=1}^{n} f(x_j, r_i) \tag{6}$$

where $\frac{a\cdot\log(n)+1}{n}$ is a score used to control the effect of the number of instances and a is a parameter which can be tuned in development. Then the relation r_i will be assigned to this pair if $lik(X, i)$ is larger than a threshold:

$$lik(X, i) > \alpha_i \tag{7}$$

where α_i is a relation-specific threshold learned from the training dataset.

[1] $\text{top\_}m(X^q)$ denotes the set of top m scoring relations for X^q under the current parameter setting.

Global Ranking for Predicted Relations

As predicted relations for different entity pairs are finally ranked together in order to get the precision/recall curve (ranking is also particularly important for high-quality extraction), it's necessary to rank the most confident ones at the highest positions. We therefore impose "global" constraints on the scoring function. Specifically, given a training dataset $\{(X^q, S^q)|q = 1, \ldots, M\}$ consisting of (mention set, relation set) pairs, the constrains are presented as follows :

$$\forall q_1, q_2, \forall r_i \in S^{q_1} \wedge r_i \notin S^{q_2}, lik(X^{q_1}, i) > lik(X^{q_2}, i) \tag{8}$$

These constraints on the likelihood of relations from different entity pairs require that for a given relation, the score of the entity pair with this relation should be larger than that of any pair without this relation. This method provides a global perspective for ranking all predicted relations, and as we will see, boosts precision significantly. The original algorithm can be easily modified to incorporate these constrains. The new algorithm is presented in Algorithm 2.

Experiments

In this section, we empirically evaluate our method and compare it with other state-of-the-art methods.

Data

We evaluate our method on the KBP dataset developed by Surdeanu et al. (2012). The KBP dataset was constructed by aligning a subset of the English Wikipedia infoboxes from a 2008 snapshot against a document collection that merged two distinct sources: (a) a collection of approximate 1.5 million documents provided by the KBP shared task (Ji et al. 2010; 2011) and (b) a complete snapshot of the English Wikipedia from June 2010. The KBP dataset contains 183,062 training gold relations and 3334 testing gold relations from 41 relation types. In practice, we use the same partition of dataset for tuning and testing as Surdeanu et al. (2012). That is , 40 queries are used for development and 160 queries are used for formal evaluation.

Baselines

We compare our method against three models:

Mintz++. This is the traditional model originally proposed by Mintz et al. (2009), yet improved to allow extracting multiple relations for an entity pair by Surdeanu et al. (2012).

Hoffmann This is a multi-instance multi-label model, which is based on At-Least-One assumption. Hoffmann uses perceptron-style additive parameter update approach during model learning.(Hoffmann et al. 2011).

MIML-RE. This is also a multi-instance multi-label model proposed by Surdeanu et al. (2012). MIML-RE is different from Hoffman in two aspects: (1) MIML-RE uses an object-level classifier to capture dependencies between relation labels; (2) it trains the model using the EM algorithm in a Bayesian framework.

Experimental Results

We evaluate our method from two different aspects. One is the overall performance of relation extraction. The other is the effectiveness of aggregating information from multiple matched instances.

Overall Results. For our method, we adopt the same feature representation for each instance in KBP as Surdeanu et al. (2012). Recall that our training dataset is represented as a set of feature vectors $\mathbf{\Phi}(x, r)$. We simply create each component of the feature vector $\mathbf{\Phi}(x, r)$ as a counting function. For example, one such component $\Phi_s(x, r)$ might be

$$\Phi_s(x,r) = \begin{cases} c & \text{if relation } r \text{ is } \textit{employee-of} \text{ and the word} \\ & \textit{company} \text{ occurs } c \text{ times in the instance } x \\ 0 & \text{otherwise} \end{cases}$$

During learning the scoring function, we only use positive training examples, and features appearing less than 5 times in positives are removed. Finally, the remaining negative training examples are added with positives to tune the relation-specific thresholds α_i.

We evaluate two variants of our approach: one (RankRE-local) that only ranks relations in the same entity pair (see Algorithm 1), and another (RankRE-global) which ranks relations across entity pairs (see Algorithm 2).

Our method has two parameters that require tuning: the number of iterations (T) and the value (a) used to control the effect of the number of instances. We tune them using the development queries, and obtain the optimal values $T = 7, a = 0.2$ for RankRE-global and $T = 11, a = 0.18$ for RankRE-local. For other baselines, we use Mintz++ , Hoffmann and MIML-RE implementation from Stanford's MIMLRE package publicly released by Surdeanu et al. (2012).

Following Surdeanu et al. (2012), we evaluate all methods using the official KBP scorer with two changes: (a) relation mentions are accepted as correct regardless of their supporting document; (b) only the subset of gold relations that have at least one mention in our sentences is scored. To compute the precision/recall curve for our method, we rank extracted relations according to the overall likelihood $lik(X, i)$. Precision/recall curves are reported in Figure 1.

Note that although RankRE-local is worse than other methods on the low-recall region of the curve, it is pretty competitive on the high-recall region (when recall is larger than 0.25). We believe that the poor performance on the low-recall region is caused by lacking ranking constraints on relations from different entity pairs. Therefore, the addition of "global" constrains in RankRE-global successfully ranks the most confident extractions at the highest positions and boosts precision on the low-recall region.

Methods	RankRE	Mintz++	Hoffmann	MIML-RE
Training time	6 minutes	3 hours	1 hour	21 hours

Table 2: Training time of all methods on the KBP dataset

Top-N	Top-50	Top-100	Top-150
Mintz++	0.540	0.410	0.333
MIML-RE	0.500	0.470	0.406
RankRE-local	**0.580**	**0.490**	**0.426**
RankRE-global	**0.560**	**0.490**	**0.433**

Table 3: Precision of top-50, top-100 and top-150 extracted relations on the dataset that contains entity pairs with at least 5 matched instances

For most regions of the curve, especially the low-recall region of the curve (when recall is less than 0.22), RankRE-global achieves significantly higher precision for the same recall point than MIML-RE. The precision improvement can be as high as 0.14 around the region of 0.05 recall. RankRE-global is comparable to, but slightly worse than, MIML-RE when recall is larger than 0.22 (the largest difference approximates to 4 precision points). This is probably because we do not use any negative example during learning the scoring function. It might produce relatively high scores for some wrong predictions. From Figure 1, we can also see that RankRE-global achieves significant improvements over Mintz++ and Hoffmann baselines. We believe these results verify that it is more appropriate to model relations which a instance explicitly or implicitly expresses by ranking than by classifying.

Another advantage of our method is the superior computational efficiency to all other baselines, as shown in Table 2. On the KBP dataset, MIML-RE requires approximately 21 hours to train; Hoffmann requires approximately 1 hour to train; the Mintz++ requires approximately 3 hours to train. In contrast, the training time of our approach is just around 6 minutes and most of the training time is spent on tuning the relation-specific thresholds α_i. The significant advantage of scalability comes from the fast parameter update of the perceptron algorithm when feature vectors in the training dataset are generally very sparse.

Multi-instance Relation Extraction Results. We test the methods described above on the subset of the original KBP testing dataset which consists of entity pairs with at least 5 matched instances. Specifically, we still use the KBP scorer in the same way, yet without extracting any relation for entity pairs associated with less than 5 matched instances. We compute the precision of top-50, top-100 and top-150 extracted relations for all methods.[2] The result is reported in Table 3.

From Table 3, we can see that both RankRE-local

[2]As the Hoffman baseline only extracts 92 relations in this experiment, we do not contain it in the result table.

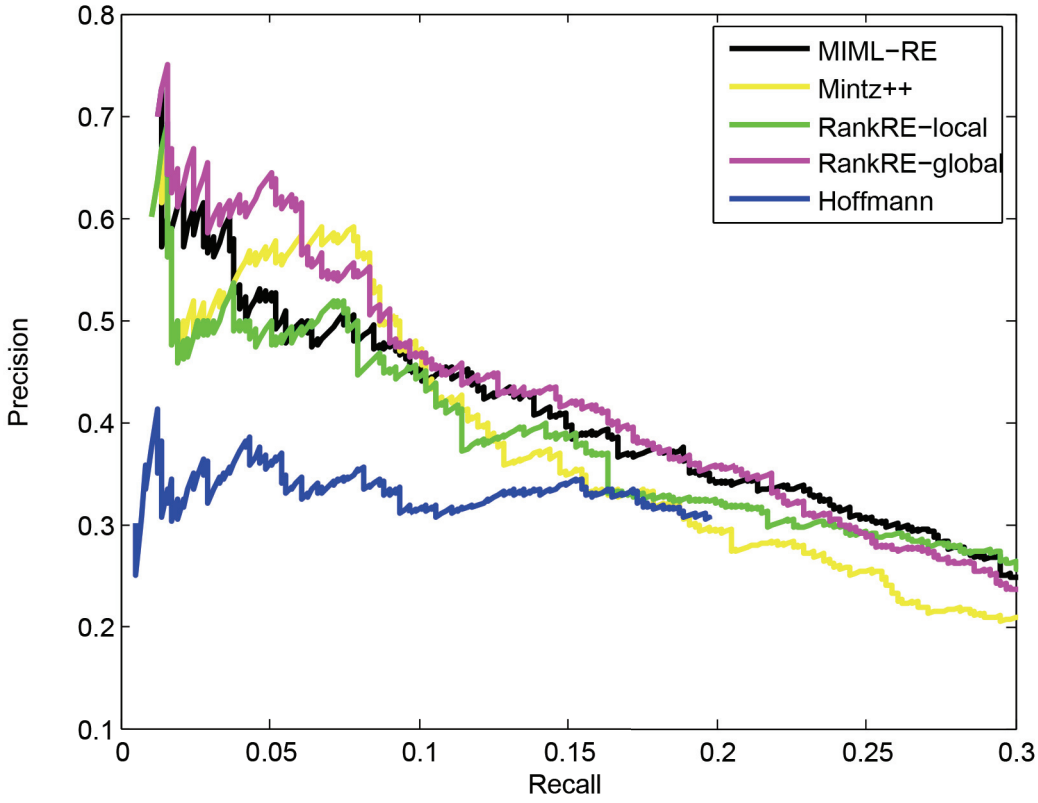

Figure 1: Precision/recall curves in KBP dataset

and RankRE-global achieve consistent improvements over Mintz++ and MIML-RE baselines. The superior performance to other baselines on "multi-instance" dataset indicates that our method can better perform aggregate entity-level relation extraction. Considering that MIML-RE explicitly deals with the multi-instance problem as well, we believe that the consistent improvement over MIML-RE proves the effectiveness of our method for exploiting inter-sentence information.

Conclusion

In this paper, we show that as relations could be expressed across multiple sentences, aggregating inter-sentence information can be leveraged to enhance distantly supervised relation extraction. With the basic idea that it makes more sense to model all possible relations in a sentence by ranking than by classifying, we propose a ranking-based method to effectively aggregate the inter-sentence information.

We evaluate our model on the large real-world dataset KBP. The results show that our approach outperforms all the state-of-the-art baselines. Our approach performs well for the overall relation extraction, and also achieves significant improvements in the multi-instance scenario which is very common in distantly supervised relation extraction.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61170189, 61370126, 61202239), National High Technology Research and Development Program of China under grant (No.2015AA016004), Major Projects of the National Social Science Fund of China under grant (No.14&ZH0036), Science and Technology Innovation Ability Promotion Project of Beijing (PXM2015-014203-000059), the Fund of the State Key Laboratory of Software Development Environment (No.SKLSDE-2015ZX-16), Disaster research funding of the people's insurance company of China. We thank the anonymous reviewers for their insightful comments.

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2787–2795.

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 89–96. ACM.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise ap-

proach. In *Proceedings of the 24th international conference on Machine learning*, 129–136. ACM.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 1–8. Association for Computational Linguistics.

Craven, M.; Kumlien, J.; et al. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, 77–86.

Freund, Y.; Iyer, R.; Schapire, R. E.; and Singer, Y. 2003. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research* 4:933–969.

Han, X., and Sun, L. 2014. Semantic consistency: A local subspace based method for distant supervised relation extraction. In *Proceedings of ACL-14, 52nd Annual Meeting of the Association for Computational Linguistics*, 718–724.

Herbrich, R.; Graepel, T.; and Obermayer, K. 1999. Support vector learning for ordinal regression.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 541–550. Association for Computational Linguistics.

Ji, H.; Grishman, R.; Dang, H. T.; Griffitt, K.; and Ellis, J. 2010. Overview of the tac 2010 knowledge base population track. In *Proceedings of the Text Analytics Conference*.

Ji, H.; Grishman, R.; Dang, H. T.; Griffitt, K.; and Ellis, J. 2011. Overview of the tac 2011 knowledge base population track. In *Proceedings of the Text Analytics Conference*.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 133–142. ACM.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 1003–1011. Association for Computational Linguistics.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 148–163.

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

Roller, R.; Agirre, E.; Soroa, A.; and Stevenson, M. 2015. Improving distant supervision using inference learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 273–278. Beijing, China: Association for Computational Linguistics.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 455–465. Association for Computational Linguistics.

Takamatsu, S.; Sato, I.; and Nakagawa, H. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 721–729. Association for Computational Linguistics.

Weston, J.; Bordes, A.; Yakhnenko, O.; and Usunier, N. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*.

Xia, F.; Liu, T.-Y.; Wang, J.; Zhang, W.; and Li, H. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, 1192–1199. ACM.

Xu, W.; Hoffmann, R.; Zhao, L.; and Grishman, R. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *ACL (2)*, 665–670.