entropy

Article

Distant Supervision for Relation Extraction with Ranking-Based Methods †

Yang Xiang *, Qingcai Chen *, Xiaolong Wang and Yang Qin

Intelligence Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China; wangxl@insun.hit.edu.cn (X.W.); yang.qin@hitsz.edu.cn (Y.Q.)

* Correspondence: xiangyang.hitsz@gmail.com (Y.X.); qingcai.chen@gmail.com (Q.C.); Tel.: +86-755-2603-3475 (Q.C.)

† This paper is an extended version of our paper published in the 22nd International Conference on Neural Information Processing, Istanbul, Turkey, 9–12 November 2015.

Academic Editor: Raúl Alcaraz Martínez
Received: 18 February 2016; Accepted: 18 May 2016; Published: 24 May 2016

Abstract: Relation extraction has benefited from distant supervision in recent years with the development of natural language processing techniques and data explosion. However, distant supervision is still greatly limited by the quality of training data, due to its natural motivation for greatly reducing the heavy cost of data annotation. In this paper, we construct an architecture called MIML-sort (Multi-instance Multi-label Learning with Sorting Strategies), which is built on the famous MIML framework. Based on MIML-sort, we propose three ranking-based methods for sample selection with which we identify relation extractors from a subset of the training data. Experiments are set up on the KBP (Knowledge Base Propagation) corpus, one of the benchmark datasets for distant supervision, which is large and noisy. Compared with previous work, the proposed methods produce considerably better results. Furthermore, the three methods together achieve the best F_1 on the official testing set, with an optimal enhancement of F_1 from 27.3% to 29.98%.

Keywords: distant supervision; relation extraction; multi-instance multi-label learning; ranking

1. Introduction

Relation extraction aims at predicting the semantic relation between a pair of name entities, such as the `govern_of` relation between a PERSON and an ORGIZATION, the `date_of_birth` relation between a DATE and a PERSON, *etc.* [1–5]. It is widely applied in knowledge bases (KB), search engines and question answering systems. Traditionally, supervised learning is widely used in this research field because the training samples annotated by human experts are high-confidence and can be directly imported into machine learning algorithms [6–9]. In recent years, with the explosive growth of information on the web, it is no longer reasonable for human experts to do extensive data annotation since new information increases exponentially and data annotation would require a great deal of time and effort. In this situation, distant supervision (DS) provides a way to alleviate this problem by aligning existing knowledge bases (KB) with batches of free text. Concretely, DS assumes that a sentence conveys the relation *r* if it contains the corresponding entity *e1* and *e2* [10–12]. Through mathematical modeling, machine learning algorithms can learn from this weak supervision (the assumption above) and build almost reliable classifiers. However, DS still faces two main challenges today: (1) we do not know in advance the exact mappings between sentences and relations; and (2) we cannot guarantee both the KB and the free text are complete.

For example, suppose an entity pair `<Barack Obama, US>` has two associated relations `president_of` and `born_in` according to a KB, and for each of them, DS assumes that *at least one* (or *all* in early studies) of the three sentences S1–S3 convey the relation (Table 1). However, we are not able to decide beforehand the actual mappings between sentences and relations (*i.e.*, S1 with `president_of`, and S2 with `born_in`). Also, due to the incompleteness problem, some sentences may express none of the relations (*i.e.*, the incompleteness of the KB leads to the conclusion that S3 conveys neither of the relations), or the sentences may not be enough to support all the relations (the incompleteness of free text).

Table 1. Examples of DS annotated instances.

	`<Obama, US>`

Relations from KB	`president_of` `<Obama, US>`
	`born_in` `<Obama, US>`

DS annotated sentences	S1. `Obama` is the 44th President of `US`. S2. Born in Honolulu, Hawaii, `US`, `Obama` is a graduate of Columbia University and Harvard Law School. S3. `Obama` talks up `US` recovery and urges Republicans to back higher wages.

To address the above issues, multi-instance learning (MIL) was proposed to model the structure of multiple instances [11,12] and further multi-instance multi-label learning (MIML) [13–15]. Different from traditional machine learning algorithms, MIL/MIML takes a *group* as the basic training unit, each of which is constituted by multiple sentences containing the same name entity pair (*i.e.*, S1–S3 in Table 1 constitute a group), and multiple sentences in the group share the same label(s) (*i.e.*, `president_of` and `born_in` for S1–S3). MIL/MIML addresses the exact mapping issue through the *at-least-one* assumption, bypassing the hard hypothesis that all sentences express the associated relation(s), and by training the instance-level classifier with the help of this assumption. MIL/MIML with *at-least-one* (weaken the incompleteness issue) and special add-ins (*i.e.*, adding penalty factors) can also partially deal with the incompleteness problem. Most previous MIL/MIML-based studies learn from all the weakly labeled training data. Differently, we train our models with a subset of the training data according to a series of ranking-based constraints, in order to tackle the incompleteness problem by filtering noise for the training data.

In this paper, we introduce MIML-sort together with three ranking-based methods so as to select effective training groups. The model is built on top of MIML-re, one of the landmark models introduced in [13]. The three methods are called MIML-sort-l, MIML-sort-p and MIML-sort-r, respectively, defining different criteria to evaluate how effective a group is (the quality of a group). Concretely, MIML-sort-l defines how the instances conform to the group-level labels, MIML-sort-p measures how representative the instances are in a group, and MIML-sort-r evaluates the correctness of the group-level labels through defining pair-wise rankings. In order to leverage the advantage of each ranking-based method, we integrate them to generate an ensemble named MIML-sort-e. Through modeling group-level and instance-level labels, we address the issue of instance-label mapping. With the ranking-based selection, we remove the noises produced due to the incompleteness of either the KB or the free text.

In the community of distantly supervised relation extraction, MIML-semi (Semi-supervised MIML) [14] and DNMAR (Distant Supervision with Data Not Missing At Random) [15] are the most closely related approaches to this work, but there are major differences. Compared with MIML-semi: (1) they only considered the missing information in the KB but not in the text; (2) they explicitly modeled the incorrectness of group-level labels through hard-coding the percentage of positive groups for each relation while we implicitly model them by means of ranking scores; (3) they changed the group-level labels before each Expectation Maximization (EM) training step while we remove those groups whose labels are likely to be wrong; (4) they tuned the parameters only to optimize the Precision-Recall curve (P-R curve) but we optimize both the P-R curve and the F_1 value on the official testing set (Final F_1). Compared with DNMAR: (1) they explicitly modeled the missing information with hard constraints but we do not; (2) they used all the groups for training but we just select those effective ones; (3) we employ hard EM for parameter estimation but they performed exact inference.

Experiments are setup on one of the landmark corpus, the KBP dataset created by [13]. We compare our approaches with MIML-semi, one of the state-of-the-art methods whose structure is mostly close to ours. Since approximate inference is more appropriate for training data large in size, we did not compare with DNMAR in this work due to its exact inference operation. Also, we report the results compared with three other baselines including MIML-re [13], Mintz++ [11] and Hoffmann [12]. We follow and define several metrics to evaluate the performance of our models (see Section 4.3). Compared with previous work, experiments show that MIML-sort-r and MIML-sort-l have considerably better results than all of them on nearly all the metrics, and MIML-sort-p produces a better Final F_1 than the baseline methods. Further, MIML-sort-e achieves the best Final F_1, with an optimal enhancement from 27.3% to 29.98%.

Our contributions can be summarized as follows:

- We are the first to make use of the group-level information to select effective training data.
- Three ranking-based methods and the ensemble are proposed and validated as effective.
- We achieve state-of-the-art results both on the average and optimal performance.
- We analyze why the data selection methods are beneficial through examples and statistical figures.

The rest of the paper is organized as follows. In Section 2 we briefly introduce the related work. In Section 3 we describe the models we proposed. Section 4 describes the experiments and analysis. Discussion, future work and conclusions are provided in the last section.

2. Related Work

2.1. Distant Supervision for Relation Extraction

Distant supervision was first introduced in the biomedical domain by mapping databases to PubMed [16], due to the existence of isomers. Since then, it has gained much attention in both information extraction (IE) and relation extraction (RE) areas. Most of earlier research including [17] and [18] used single label learning, meaning they believed sentences containing some specific marks must express certain meanings. In recent years, distant supervision has been widely used in open IE (*i.e.*, mapping Wikipedia info-boxes to wiki contents or web scale text) [19], which is a good way to automatically generate info-boxes for new articles. For RE, distant supervision was employed on mapping relations from KBs (*i.e.*, Freebase) to a large scale of text (*i.e.*, New York Times or electronic magazines) and predicting relations for unseen entity pairs according to sentences [10,11]. However, this assumption does not make sense in many cases since sentences can convey multiple meanings when containing identical marks. To address this issue, MIL/MIML was later proposed, which transforms the data into styles with multiple sentences and single/multiple labels. Concretely, Hoffmann *et al.* [12] and Surdeanu *et al.* [13] proposed MIML to enable overlapping relations. References [14,15] are extensions to [13] with additional layers or add-in penalty factors. Recent work also included the embedding models that transfer the relation extraction problem into a translation model like $h + r \approx t$ [20–22], and the probability matrix factorization (PMF) models from [11,23] in which training and testing are carried out jointly. Besides, Fan *et al.* [24] presented a novel framework by integrating active learning and weakly supervised learning. Nagesh *et al.* [25] solved the label assigning problem with integer linear programming (ILP) and improved the baselines. In addition, there are some deep learning based methods using convolutional neural networks to do feature modeling and MIL to do distant supervision [26]. Although different learning architectures have been proposed, how to efficiently make use of the noisy training data is still the main challenge for distant supervision.

2.2. Noise Reduction for Distant Supervision

As mentioned previously, noise is easily imported due to the unknown mapping between relations and sentences, and also due to the incompleteness problem. Many studies focused on methods on reducing noise for distant supervision. Several *at-least-one* learners (MIL/MIML) were proposed which consider the relation label to be positive in the group when at least one of the sentences in the group conveys the relation [11–13]. Intxaurrondo *et al.* [27] proposed several heuristic strategies to remove useless mentions but mainly on the instance-level, *i.e.*, clustering the instances and keeping the central ones. Xu *et al.* [28] employed a passage retrieval model to expand the training data based on a small set of seeds. Min *et al.* [14] discarded the generated labels but added another layer to the MIML-re framework to model the true labels of a group. Takamatsu *et al.* [29] directly modeled the patterns that express the same relation by using co-occurrence. Ritter *et al.* [15] added two penalty factors to model the missing of texts and the missing of KB, and they also considered some additional information such as popularity of entities. Angeli *et al.* [23] added a bias factor b to their PMF algorithm to model the noise. Xiang *et al.* [30] computed the value of two types of biases to model the correctness and incorrectness for each group-level label. Our early work proposed two ranking-based group selection methods and achieved promising results [31]. However, most the above work is built on the idea that the *at-least-one* assumption can be satisfied, which does not consider the incompleteness problem.

Among the previous studies, references [14,15] are typical work that take the incompleteness problem into consideration. However, MIML-semi [14] only considered the incompleteness of the KB, and it hard-encoded the percentage of positive instances in groups, which is difficult to extend to other datasets. DNMAR [15] also faced the problem by setting fixed penalties for missing, and the exact inference algorithm had high computational requirements.

MIML-sort, introduced in this paper, tackles the mapping problem by using the MIML architecture, and deals with the incompleteness problem by adding soft penalties, utilizing the characteristics of the groups themselves.

3. Methods

3.1. MIML-Sort

We built our models on top of MIML-re, the most widely used multi-instance multi-label learning framework for relation extraction. MIML-re places multiple sentences that contain the same entity pair into a group (multi-instance), and enables the group members to share all existing relation labels for this pair (multi-label). A two-level classifier is constructed to learn the latent sentential labels using all the positive training groups (with heuristically generated labels) and a subsample of the negative groups (without labels). The primary heuristic (assumption) that MIML-re follows is *at-least-one*: at least one sentence in a group conveys any group level relation. We show the plate model of MIML-re in Figure 1 and introduce the model below.

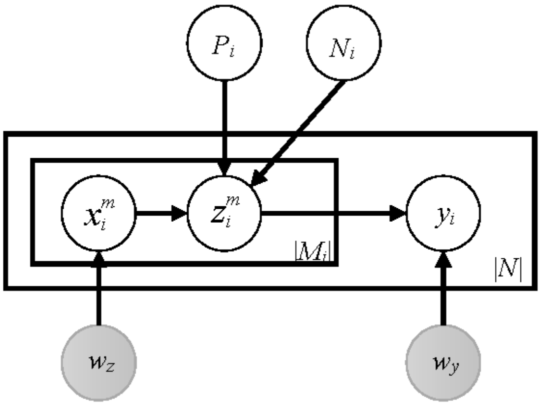

Figure 1. The plate model of MIML-re. (The smaller rectangle denotes the sentence-level classifier, with number of repetitions $|M_i|$—the number of sentences in group i. The larger rectangle is the group-level classifier, repeated $|N|$ times—the number of training groups.)

The architecture of MIML-sort is analogous to MIML-re except that it ranks the training groups at the end of each training epoch according to corresponding scoring functions, and uses a subset of training data for the next training epoch. In this section, we subtly introduce MIML-sort and its differences from MIML-re. For a clearer description, we use the term *sort* to denote the ranking of groups, and the term *rank* to denote the ranking of instances (within a group).

Following MIML-re, MIML-sort defines positive labels (P_i) as all the possible relation labels from the KB, while negative labels (N_i) are defined as $ŁP_i$, where $Ł$ stands for the possible relation labels for the key entity (the first entity in the pair). For instance, `president_of` and `born_in` are included in P_i for the entity Obama in the example shown in Table 1, while some other relations such as `founder_of` can be components for N_i. The joint probability of the whole dataset D is defined as:

$$p(D) = \prod_{i=1}^{N} p(y_i|x_i, \boldsymbol{w}_y, \boldsymbol{w}_z) = \prod_{i=1}^{N} p(y_i, z_i|x_i, \boldsymbol{w}_y, \boldsymbol{w}_z) \tag{1}$$

$$p(y_i|x_i, \boldsymbol{w}_y, \boldsymbol{w}_z) = \prod_{m=1}^{M_i} p(z_i^m|\boldsymbol{w}_z) \times \prod_{r \in P_i \cup N_i} p(y_i^r, \Big| z_i, \boldsymbol{w}_y^r) \tag{2}$$

The parameters of the model include the instance-level classifier $\boldsymbol{w}_z$ and the group-level classifier $\boldsymbol{w}_y$. $\boldsymbol{w}_z$ is a multi-class classifier which maps each instance to one of the relation labels. $\boldsymbol{w}_y$ is a series of binary classifiers which utilize the information from the predicted labels from $\boldsymbol{w}_z$. In Equations (1) and (2), x_i and z_i stands for the instances (sentences) and the corresponding predicted labels for them in Group i, while y_i represents the group-level labels for the group. We use z_i^m to denote the m-th sentence in Group i, y_i^r the r-th label, and $\boldsymbol{w}_y^r$ the binary group-level classifier for label r.

Following MIML-re, we use the hard EM algorithm to estimate the parameters $\boldsymbol{w}_y$ and $\boldsymbol{w}_z$ through several training epochs. In the E-step, the algorithm traverses each instance in each group, and uses $\boldsymbol{w}_z$ and $\boldsymbol{w}_y$ from the last training epoch for prediction. The predicted label for each instance is generated by maximizing the item on the right side in Equation (3).

E-step:

$$z_m = \arg\max_z p(z|x_i^m, \boldsymbol{w}_y, \boldsymbol{w}_z) = \arg\max_z p(z|x_i^m, \boldsymbol{w}_z) \times \prod_{r \in P_i \cup N_i} p(y_i \Big| z_i\prime, \boldsymbol{w}_y^r) \tag{3}$$

where $z_i{}'$ denotes the group labels inferred previously in which the m-th label z_m is replaced by the current candidate label z, and x_i^m stands for the mth instance in the i-th group. Rewriting in log form:

$$z_m = \arg\max_z \log p(z|x_i^m, \boldsymbol{w}_y, \boldsymbol{w}_z) = \arg\max_z [\log p(z|x, \boldsymbol{w}_z) + \sum_{r \in P_i \cup N_i} p(y_i \Big| z_i\prime, \boldsymbol{w}_y^r)] \tag{4}$$

We perform the sorting operation (S-step) following each E-step after each instance has been assigned with the optimal label, and the S-step is the main difference from MIML-re. We use *f*(x_i) to denote the score of each group, which is further defined based on different sorting strategies, and the training dataset is updated by selecting the top θ groups according to *f*(x_i) (see Equations (5) and (6)).

S-step:

$$X_{e+1} = ranking(X_e) \downarrow, s.t. \forall i, j,\ if(i < j),\ then\, f(x_i) > f(x_j) \tag{5}$$

$$D_{e+1} = \{x_i\}\ , s.t.\ i \leqslant \theta \cdot |X_{e+1}| \tag{6}$$

where D_{e+1} is the training dataset for the (e + 1)-th epoch and X_{e+1} is the universal set of training groups in the (e + 1)-th epoch.

The parameters of the model are maximized after the S-step through Equations (7) and (8) in the M-step. The updated training dataset D_{e+1} is generated by Equations (5) and (6).

M-step:

$$\boldsymbol{w}_z^* = \arg\max_{w} \sum_{i=1}^{N} \sum_{m \in M_i} \log p(z_i^m | x_i^m, \boldsymbol{w}, D_{e+1}) \tag{7}$$

$$\boldsymbol{w}_y^{r*} = \arg\max_{w} \sum_{1 \leqslant i \leqslant N, r \in P_i \cup N_i} \log p(y_i^r | z_i^*, \boldsymbol{w}, D_{e+1}) \tag{8}$$

where $\boldsymbol{w}_y^{r*}$ stands for the optimal group-level classifier for relation r. Equations (9) and (10) are used to infer the instance-level and group-level labels, respectively.

Inference:

$$z_i^{m*} = \arg\max_{z} p(z | x_i^m, \boldsymbol{w}_z) \tag{9}$$

$$y_i^r = \arg\max_{\{0,1\}} p(y | z_i^*, \boldsymbol{w}_y^r) \tag{10}$$

where z_i^{m*} stands for the predicted relation label for the m-th sentence in the ith bag, and y_i^r is the binary selection for the rth relation label for Group i.

Algorithm 1 MIML-Sort Training

```
Input: training bags $\{x_i\}$, positive/negative label sets $\{P_i/N_i\}$, label set $R$, proportion parameter $\theta$
Output: instance-level and group-level classifiers $\boldsymbol{w}_z$ and $\boldsymbol{w}_y$
foreach $x_i^m$ in each bag $x_i$:
    $z_i^m \leftarrow$ each $r$ in $P_i$
end for
foreach iteration $t$ in $T$:
    foreach bag $x_i$:
        foreach $x_i^m$ in each bag $x_i$:
            $z_m = \arg\max_{z} p(z | x_i^m, \boldsymbol{w}_y, \boldsymbol{w}_z) = \arg\max_{z} p(z | x_i^m, \boldsymbol{w}_z) \times \prod_{r \in P_i \cup N_i} p(y_i = r \Big| z_i', \boldsymbol{w}_y^r)$
        end for
    end for
    foreach bag $x_i$:
        $z_i* = \arg\max_{z} p(z | x_i, \boldsymbol{w}_z)$
        foreach $r$ in $R$:
            $y^{r*} = \arg\max_{\{0,1\}} p(y | \mathbf{z}_i^*, \boldsymbol{w}_y^r)$
        end for
    end for
    $D_{e+1} \leftarrow \delta(D_e, \theta),\ given\ \{f(x_i)\}$
    $\boldsymbol{w}_z^* = \arg\max_{w} \sum_{i=1}^{n} \sum_{m \in M_i} \log p(l_i^{m*} | x_i^m, \boldsymbol{w}, D_{e+1})$
    foreach $r$ in $R$:
        $\boldsymbol{w}_y^{l*} = \arg\max_{w} \sum_{i,l} \log p(y_i^l \Big| z_i^*, \boldsymbol{w}, D_{e+1})$
    end for
end for
```

In the testing phase, MIML-sort also uses a *noisy-or* model analogous to MIML-re to avoid data sparsity and gain better results.

Noisy-or:

$$p_i(r | \boldsymbol{w}_z) = 1 - \prod_{m \in M_i} [1 - p(r | x_i^m, \boldsymbol{w}_z)] \tag{11}$$

The noisy-or model is employed within a group i by considering the predicted scores for relation r for all the instances associated with the entity pair. This formula integrates not only the model confidence (by multiplying the predicted score on relation r for each instance) but also the redundancy information (by taking the number of instances into account).

The description of the training process is shown in Algorithm 1. In the algorithm, Lines 1–3 are the initialization process, with the operation that each label in P_i is assigned to each instance, and a set of instances of size $|M_i| \times |P_i|$ is generated. The rest lines are similar to MIML-re except for an updated version of dataset D_{e+1} shown in Lines 16–19. The model parameters $\boldsymbol{w}_y$ and $\boldsymbol{w}_z$ are thus estimated through several iterations.

Following this, we introduce three types of sorting methods based on different strategies. Then, to leverage their advantages, we combine them into an ensemble. The motivation for us to sort and select is that we expect to reduce the noisy part in the training data so that the negative effect of the noisy data can be reduced as much as possible. In our view, an ideal group for distant supervision is one that contains a *perfect mapping*: each group-level label can be mapped to at least one sentence, and the relation expressed by each sentence can be found in the group-level positive labels, which is difficult to achieve just relying on the weakly supervised data annotation. Meanwhile, we expect the redundant information in a group to be minimized (*i.e.*, useless sentences or labels).

3.2. Sorting by Conformance of Group-Level Labels

As mentioned in the previous section, each training group is constituted by a set of instances containing the same entity pair (E_i), a set of positive candidate relation labels that all instances share (P_i), and a set of negative relation labels that this entity pair does not express (N_i). In this method, we assign a score to each group by summing up the *sub-score*s for all positive labels and negative labels for the group.

We define K_l as the key instance for relation l—the instance that has the maximum predicted score for a certain label in the group.

$$K_l = \arg\max_m p(z_m = l|\boldsymbol{w}_z, x_i^m) \tag{12}$$

For each l in P_i, the sub-score is the predicted confidence for this label, while for each l' in $N_{i,}$ the sub-score is the confidence in not predicting for the label. For each label in P_i and N_i, the sub-scores are selected from the key instances correspondingly. The final score for group i is defined as:

$$f(x_i) = \frac{1}{P} log \sum_{r \in P_i} p(K_r) + \frac{1}{N} log \sum_{\bar{r} \in N_i} [1 - p(K_{\bar{r}})] \tag{13}$$

where Z_p and Z_N are normalization factors which are set to be $|P_i|$ and $|N_i|$. This sorting strategy is inspired by the original MIML-re model in which the label assignment for an instance is partly determined by the group level generative probabilities.

The score reinforces the certainty for a group to the group-level labels. It can be seen as a form of negative entropy of a group. The higher the score is, the lower the entropy is, and the more effective a group is. The motivation for proposing this score is to measure how the instances (sentences) conform to the group-level labels and what we expect to retain are the groups with higher scores. In an extreme case where the sentences in the group convey none of the group-level relations, which could be produced when the text is far from being complete, the probability for each group-level label should be a very small value, leading to a small *f*($\boldsymbol{x}_i$). This means the method can implicitly model the missing of the text.

To illustrate, we take the example from Table 1 as a case study. Supposing S1 did not exist, the sentence-level and the group-level probability (see the right part in Equation (3)) would both be very small, and the selecting algorithm is inclined to reject such incomplete training groups.

3.3. Sorting by Precision of Labels

This sorting method is inspired by [27] in which the authors selected the instances with high quality through clustering. We believe that selecting instances that are unambiguous is beneficial in most cases for machine learning algorithms. Analogous to *noisy-or* (Equation (10)), we define an equation by replacing l with the predicted label r* for each instance. The label with this confidence can be easily obtained from any classifier (*i.e.*, Linear Regression in MIML-sort).

$$r* = \arg\max_{r} p(z_m = r|\boldsymbol{w}_z, x_i^m); m \in M_i \tag{14}$$

The score for group i is defined as:

$$f(x_i) = 1 - \frac{1}{} \sum_{m \in M_i} [1 - p(z_m = r*|\boldsymbol{w}_z, x_m)] \tag{15}$$

where Z is the normalization factor and is set to be the size of the group $|M_i|$.

The predicted confidence for a certain label reflects the distance from an instance to the corresponding class center. It is obvious that the smaller this distance is, the more possible the instance belongs to this class. Since the training corpus is automatically generated without any human intervention, the quality of the instances can hardly be guaranteed. For example, the sentence `Obama was born and grew up in the US.` may be either classified into the relations `born_in` or `resident_in`, which adds confusions to classifiers. However, retaining the high quality instances may cause the local minimum problem (solved by the N-fold bagging strategy mentioned later) since the proportion of the sentence-level probability can be too large compared with the group-level probability.

3.4. Sorting by Ranking of Group-Level Labels

We define the term *ranking* according to the relative order of two labels in a group. More specifically, within a group, if the predicted score of label l is larger than that of label l', we say l has a higher rank than l' (here the predicted score for label l is from the key instance). For easier description, we define the following notations:

R_l—the number of labels that has a higher predicted score than label l within a group.
L_l—the ranking loss for a certain label l in a group which is related to R_l.

We use the above ranking loss to model the case when at least one non-positive label ranks higher than a positive label.

More formally, we define:

$$R_l = \sum_{l\prime \notin P_i} I[p(K_{l\prime}) > p(K_l)] \tag{16}$$

where $I[\cdot]$ is an indicator function which returns 1 if the inner argument is true and 0 otherwise, and

$$f(x_i) = -\frac{1}{} \sum_{l \in P_i} L_l = -\frac{1}{_1} \sum_{l \in P_i} \frac{1}{_2} \sum_{t=1}^{R_l} 1/t \tag{17}$$

where Z_1 and Z_2 are the normalization factors which are finally set to be $|P_i|$ and $|U_i|$, respectively. Here, $|U_i|$ denotes the number of non-positive labels that has supportive instances (instances that classified into this class label) in the group. The series $\sum_{t=1}^{R_l} 1/t$ is incremental, so that any label l that has a larger R would result in a larger loss, indicating that this label is inclined to be negative in this group. This sorting strategy is inspired by [32–34] where they define a similar ranking loss to train multiclass classifiers based on pair-wise structures.

As we know, wrong positive/negative labels are inevitable due to the DS heuristics (the data annotation), and this sorting method aims at removing those groups that are more likely to contain the wrong labels. Intuitively, if a group has more positive labels recognized as "low-ranking", that group is likely to be less effective. If the KB is incomplete and there are quite a number of instances for non-positive labels, a positive label is likely to be ranked after several non-positive ones, which can be reflected by a low $f(\mathbf{x}_i)$. If the text is incomplete and one positive label can be mapped to none of the instances, the score may also be very low. Consequently, this sorting method can implicitly reflect the missing of either the KB or the text.

For a clearer description, we also take the entity pair in Table 1 as an example. Suppose S1 is missing from the sentences, the relation `president_of` would be ranked after some non-positive labels such as `talk_up(S3)`, and if `president_of` is missing, S1, which may be correlated to a non-positive label at this time, is likely to have a higher rank than some positive labels.

3.5. Sorting by Ensemble

It is precisely the different emphases (*i.e.*, modeling the missing in the KB or the text) on the above sorting strategies that enable us to combine them together and generate an ensemble model. In this paper, the three sorting methods are integrated linearly with pre-assigned weights:

$$f_e(x_i) = (\alpha, \beta, \gamma) \cdot [f_l(x_i), f_p(x_i), f_r(x_i)]^T \tag{18}$$

where α, β, γ are weight factors and the fs right-handed denote the normalized scores generated by the three sorting methods (l for conformance, p for precision, and r for ranking). We use f_e to denote the score used in the ensemble.

Of course there are also other methods to integrate the different strategies, such as learning to rank or majority voting based on the final predicted results, but here we just simply make a linear combination, to simplify the model architecture and reduce the time cost.

4. Experiments and Analysis

4.1. Dataset

We did testing on the KBP dataset, one of the landmark datasets for distantly supervised relation extraction constructed by [13] (http://nlp.stanford.edu/software/mimlre.shtml). The resources of this corpus are mainly from the TAC KBP 2010 and 2011 shared tasks [35,36]. It contains 183,062 training entity pairs and 3334 testing pairs. The free text provided by the shared task contains approximately 1.5 million documents from a variety of sources, including newswire, blogs and telephone conversation transcripts. The KB is a snapshot of the Wikipedia dump of the English version. After the DS annotation and a subsample of 5% data as negative (following MIML-re), we finally got 524,777 groups including 950,102 instances. We used the above groups as the training set. The evaluation set contains 200 queries (one query refers to a key entity) from the testing data of the shared tasks, which include 23 thousand instances. Following previous work, 40 out of 200 testing queries are the developing set (for parameter optimization) and the remaining 160 are the testing set.

4.2. Implementation

4.2.1. Implementation Details

Other than directly removing the potentially less effective groups, we used the selected top θ groups to train the linear classifiers but ranked with all groups for selection. Under this situation, some groups that have been removed in earlier epochs may be again selected according to an updated model.

For comparison, we set the maximum number of EM iterations to 8 (also with one epoch for initialization) and used a 3-fold bagging strategy to avoid over-fitting. We fixed the negative samples and removed other random settings for the sake of fair comparisons in the experiment.

We follow the settings in MIML-re for the instance-level classifier: we set Logistic Regression as the base linear classifier and used the features extracted by [13] including entity types, dependency relations, words, part-of-speeches and distance between the entity pair, *etc.* More details can be seen in [13].

For the sorting methods proposed, we tuned the parameters θ and T (number of *epoch*s) on the developing set and used the tuned value on the testing set. Finally, we got (θ, T) = (0.98, 6) for MIML-sort-l, (0.99, 8) for MIML-sort-p, and (0.98, 2) for MIML-sort-r. For MIML-sort-e, we used (0.98, 7) and (α, β, γ) = (0.4, 0.2, 0.4).

4.2.2. Baseline Methods

We implemented MIML-semi [14], one of the state-of-the-art works in this field, and compared it with our four models. Following previous research, we also compared our models with the classical baseline MIML-re and two other landmark algorithms: (1) *MultiR* (denoted as *Hoffmann* in the experiment) [12], which is a multi-instance learning algorithm that supports overlapping relations; (2) *Mintz++*, Surdeanu *et al.*'s [13] implementation of the *Mintz* model [10], which performs very well as MIML-re in some cases. We called our ensemble model MIML-sort-e in the following experiments.

4.3. Evaluation Metrics

- P/R curve

Following previous work, we report the Precision-Recall curve (P-R curve) to test the models' stability on the testing data. A P-R curve is generated through computing precision and recall by selecting different proportions of the testing results. Generally speaking, if one curve is located on top of another, the corresponding method is better.

- Precision, Recall, Final F_1

These three metrics are evaluated to see the final performance on the official testing set. Final F_1 is usually the main performance measure. We tuned the parameters on the development set to maximize Final F_1 analogous to MIML-re [13].

- Max F_1 & Avg F_1

Max F_1 means the maximum F_1 point on the P-R curve. The P-R curve shows the average performance of an algorithm, and can also be reflected by average (Avg) P/R/F_1 values.

In addition, the time cost (complexity) of the methods can be partly reflected in the parameter T and θ. Generally, the smaller T or θ is, the less complex the method is.

4.4. Results

The P-R curves are shown in Figures 2 and 3 and the detailed figures are listed in Table 2.

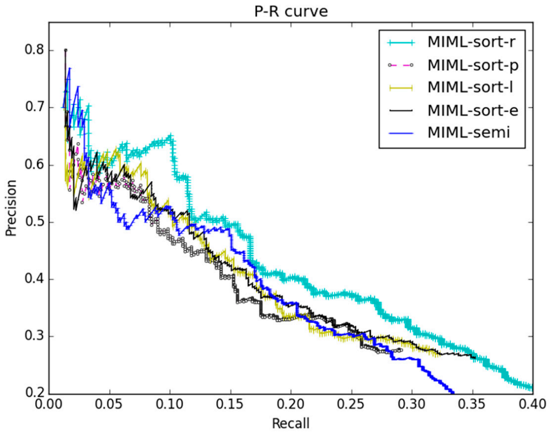

Figure 2. P-R curves for MIML-semi and our methods.

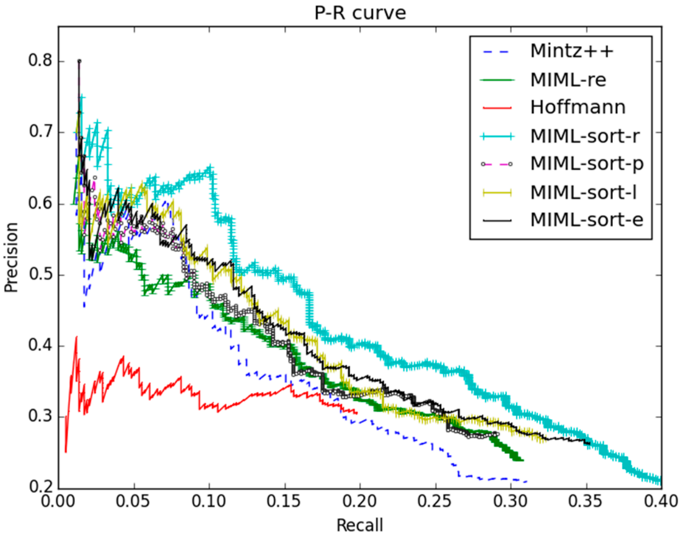

Figure 3. P-R curves for three baselines and our methods.

Table 2. Results of the proposed methods and baselines.

	Precision	**Recall**	**Final F_1**	**Max F_1**	**Avg F_1**	**Parameters**
Hoffmann	**30.65** [1]	19.79	23.97	24.05	15.40	-
Mintz++	26.24	24.83	24.97	25.51	21.61	-
MIML-re	30.56	24.68	27.30	28.25	22.75	T [2] = 8
MIML-semi	13.38	**42.88**	20.39	28.28	23.27	$T = 8$
MIML-sort-l	27.00	32.29	29.41	29.55	23.33	$T = 6$, $\theta = 98\%$
MIML-sort-p	27.50	29.17	28.31	28.33	22.05	$T = 8$, $\theta = 99\%$
MIML-sort-r	20.95	39.93	27.48	**31.29**	**26.87**	$T = 2$, $\theta = 98\%$
MIML-sort-e	26.09	35.24	**29.98**	30.32	24.34	$T = 7$, $\theta = 98\%$

[1] The optimal result for a column is marked in bold; [2] T includes the 1st epoch for initialization.

4.4.1. Compared with MIML-Semi

We can see from the P-R curves in Figure 2 that MIML-sort-p and MIML-sort-l outperform MIML-semi in the recall area (0.05, 0.1) and (> 0.3) but are inferior to MIML-semi in the area of (0.1, 0.3). MIML-sort-e has roughly similar performance to MIML-semi along the low recall curve and outperforms it when recall > 0.2. It is obvious that MIML-sort-r significantly exceeds MIML-semi in almost all recall areas, as observed from the curves. MIML-sort-r, MIML-sort-e, and MIML-sort-l have larger upper bonds on recall than MIML-semi (precision of MIML-semi drops to 0 before r = 0.35).

From the figures in Table 2 we can see that although MIML-semi has a good P-R curve, the performance on the Final F_1 is far from satisfactory, due to the very low precision. This is probablydue to that the target for MIML-semi is to tune the P-R curve but not the Final F_1. However, our methods not only gain good Avg F_1 (Avg F_1 of MIML-sort-r is 3.32% higher than that of MIML-semi) but also good Final F_1 (far better than MIML-semi). We also notice that the Max F_1 values of MIML-sort(s) are all higher than MIML-semi, with MIML-sort-r the best of all, 3.1% better than MIML-semi.

To conclude, MIML-sort-e is better in terms of precision and performs much more stable than MIML-semi, which can be seen from the P-R curve. We believe the label ranking strategy is quite helpful for distant supervision in choosing training groups. The stability of other MIML-sort systems is comparable with MIML-semi, but they all significantly outperform MIML-semi on Final F_1. The results show the effectiveness of the selection strategies. The best Final F_1, which is naturally achieved by MIML-sort-e, shows that the ensemble absorbs the advantages of the base methods.

4.4.2. Compared with Other Baselines

It is clear that MIML-sort(s) behave beyond the three baselines. Among the methods, MIML-sort-l always performs better, especially in the low recall region (< 0.1). MIML-sort-p also maintains higher precision than MIML-re in the low recall region, while being higher than Mintz++ in the high recall region. MIML-sort-r significantly outperforms all the baselines, and its precision is even 10% higher than MIML-re and Mintz++ around the recall = 0.1 point. It also extends the upper bond of recall to 0.4 which is a major improvement over other work. The curve of MIML-sort-e is also remarkable compared with baselines, and it reaches the highest precision when the recall is very low and still maintains a relatively high precision when recall > 0.3.

Some improvements can be seen more clearly in Max F_1 and Avg F_1 in Table 2. MIML-sort(s) have better Max F_1 than all other work. Avg F_1 directly reflects the level of the P-R curve, and we find that MIML-sort-r's figure is prominent, which is 4.12% higher than MIML-re and 5.26% higher than Mintz++. The official test results of KBP slot filling evaluation can be recognized from precision, recall and Final F_1. We notice that MIML-sort-e gains the best Final F_1, with an enhancement of 2.68% than MIML-re and 9.59% than MIML-semi. Other MIML-sort(s) also outperform the baselines. In addition, we see that our methods mostly benefit from recall (an exception is MIML-semi who has a very imbalanced precision and recall). We think this is mainly due to the improvement of data quality by applying the ranking-based methods, which reduces ambiguities for classification.

Another set of information shown in Table 2 are the parameters when the algorithm converges to the optimal. It is noticed that most Ts for MIML-sort(s) are less than 8 which is proven to be optimal in MIML-re. This indicates that our proposed methods can make the EM algorithm converge faster thereby lower down the computational cost. Particularly, the optimal T for MIML-sort-r is only 2, which saves a lot of running time. The values for proportion parameter θ show that we did not remove too many training groups and most of the heuristically generated data were useful for the MIML training process. More details about the characteristics of the removed groups are described in the discussion section.

5. Discussion and Future Work

We see that removing the less effective training groups boosts the system's recall but maintains good precision so that F_1 can be improved. Comparing the three basic methods we propose, we learn that MIML-sort-r performs best (from the curve), in terms of both the maximum and the average performance. The ranking strategy used in this method is validated as very effective for distantly supervised relation extraction and we think it also can be integrated in other learning frameworks, behaving as an element for the loss function. Compared with MIML-sort-r, MIML-sort-l and MIML-sort-p both have their drawbacks: MIML-sort-l does not consider the imbalanced distributions for instances (*i.e.*, a sentence with a probability of 0.8 might be strongly classified to relation l but only weakly to relation l') and MIML-sort-p lacks the information of group-level labels. However, through computing rankings between relations, MIML-sort-r tends to retain the groups that are most likely to be correctly annotated.

We also believe the extension to the upper bound of recall on the curve is a remarkable issue through which we can make more flexible adjustments to the results according to different applications. In this section, we further discuss in which case a group is likely to be removed.

5.1. Analysis of the Removed Groups

We analyzed the characteristics of the removed groups by recording the number of instances in them before the last training epoch (see Table 3 for details).

Table 3. The statistics for the removed data. The figures show the percentage of the removed groups within certain columns.

	S = 1 (%)	**S = 2 (%)**	**S = 3 (%)**	**S = 4 (%)**	**S ⩾ 5 (%)**
MIML-sort-l	46.10	25.99	11.94	6.73	9.24
(sum)	(46.10)	(72.09)	(84.03)	(90.76)	(100)
MIML-sort-p	94.63	4.54	0.65	0.11	≈0
(sum)	(94.63)	(99.16)	(99.81)	(99.92)	(100)
MIML-sort-r	75.53	12.48	5.07	2.44	4.48
(sum)	(75.53)	(88.01)	(93.08)	(95.52)	(100)
MIML-sort-e	86.32	11.32	1.48	0.50	0.38
(sum)	(86.32)	(97.64)	(99.11)	(99.62)	(100)

In the table, the title row signifies the numbers of instances contained in the removed groups (*i.e.*, S = 1 stands for the singleton groups and S ⩾ 5 for the groups which contain more than 5 instances). Each field for a certain method is constituted by two rows, in which the first row shows the percentage of the corresponding removed groups, and the second row shows the accumulation so far. For example, we can see from Table 3 that 25.99 in the 2nd row and 3rd column indicates that 25.99% of the removed groups contain 2 instances for MIML-sort-l, and 72.09 in the 3rd row and 3rd column (in brackets) shows that 72.09% of the removed groups contain fewer than 3 instances. It is easy to discover a trend that the groups with fewer instances are more likely to be removed. As well, we notice that the groups that contain 1-4 instances account for over 90% of the removed training groups. We further discover that MIML-sort(s) did not remove any group containing over 10 instances. We listed in Table 4 the data distribution in the original training data in a similar way as in Table 3.

Table 4. The distribution on group size in the original training data (with S ⩾ 10 (1.8%)).

	S = 1 (%)	**S = 2 (%)**	**S = 3 (%)**	**S = 4 (%)**	**S ⩾ 5 (%)**
Percentage	73.15	12.69	5.66	2.85	6.25
(sum)	(73.15)	(85.84)	(90.91)	(93.76)	(100)

We can read from Tables 3 and 4 that the filtered data for MIML-sort-r has a similar distribution as the original dataset, indicating that for small groups (S < 5), it is possible for them to either be effective or not, depending little on the sorting method. However, it seems that MIML-sort-p inclines to remove singletons but MIML-sort-l tends to remove larger groups. We believe that the characteristics of the removed data depend heavily on the data themselves. It is probable that the instances in the singleton groups are more ambiguous (corresponding to MIML-sort-p) but most of them can still be classified into the group-level labels (corresponding to MIML-sort-l).

Perhaps the reason why groups that have multiple instances (*i.e.*, ⩾ 10) are more effective is that the missing of data (KB/text) problem are more likely to be avoided for these more *popular* entity pairs. Ritter *et al.* [15] validated this assumption in which they assigned different penalties according to the popularity of entities.

5.2. Parameter Settings

Here we discuss T. With a subset of the training data, the label assignment for instances, which concerns both the instance-level and group-level predictions, can be quite distinct. This means that improving the quality of the training data can directly boost the algorithm's running speed, due to both the smaller training size and faster convergence of the linear classifiers (*i.e.*, $T = 2$ makes the algorithm converge to the maximum for MIML-sort-r). However, when we traversed each step on T (up to 8) and used different parameters for testing, we found it interesting that very similar maximum points distributed over several epochs. This is also a point worthy of further research.

5.3. Analysis on Relation Types

We compare the results generated by MIML-re and MIML-sort-e (the ensemble that generates the best Final F_1) on the official testing set in order to see if the sorting method boosts the baseline only on some particular relations. The results show that although MIML-sort-e has a significant enhancement on some certain relations such as `per:title` (correctly tagged from 14 to 37 and the proportion of this relation is also large), but it also has considerable improvements on other relations such as `per:parents`, `org:member_of` and `per:origin`. Therefore, we can say that the sorting strategy is useful among different relation types.

There are also some points that have to be further developed. For example, more intelligent methods need to be studied to learn the parameters automatically and quickly (*i.e.*, [34]). Also, we believe that we can automatically evaluate the importance of the three basic sorting methods through searching optimally when implementing the ensemble with some ranking algorithms.

6. Conclusions

In this paper, we propose three ranking-based methods according to different strategies in order to select effective training groups for multi-instance multi-label learning. Finally, the three methods are combined linearly to generate an ensemble. Experiments compared with baselines validate the efficiency of the proposed ranking-based methods. Particularly, MIML-sort-e boosts the F_1 value on the official testing set significantly by 2.68%, and the other MIML-sort(s) methods also produce considerable improvements from the baselines. We believe that the proposed ranking strategies can be integrated in other learning frameworks. From the results we notice there is still plenty of room for improvements, demanding more efficient and robust methods.

Acknowledgments: This work was supported in part by National 863 Program of China (2015AA015405), NSFCs (National Natural Science Foundation of China) (61402128, 61473101, and 61272383), Strategic Emerging Industry Development Special Funds of Shenzhen (JCYJ20140417172417105, JCYJ20140508161040764 and JCYJ20140627163809422), Innovation Fund of Harbin Institute of Technology (HIT.NSRIF.2017052) and Program from the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (93K172016K12).

Author Contributions: Yang Xiang and Qingcai Chen conceived and designed the experiments; Yang Xiang performed the experiments; Yang Qin analyzed the data and contributed statistics; Yang Xiang wrote the paper; Xiaolong Wang checked and modified the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Miller, S.; Fox, H.; Ramshaw, L.; Weischedel, R. A Novel Use of Statistical Parsing to Extract Information from Text. In Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL), San Diego, CA, USA, 29 April 2000; pp. 226–233.
2. Collins, M.; Duffy, N. Convolution Kernels for Natural Language. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 3–8 December 2001; pp. 625–632.
3. Zelenko, D.; Aone, C.; Richardella, A. Kernel Methods for Relation Extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
4. Kambhatla, N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Barcelona, Spain, 21–26 July 2004.
5. Culotta, A.; Sorensen, J. Dependency Tree Kernels for Relation Extraction. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Barcelona, Spain, 21–26 July 2004.
6. Bunescu, R.C.; Mooney, R.J. A Shortest Path Dependency Kernel for Relation Extraction. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, BC, Canada, 6–8 October 2005; pp. 724–731.

7. Zhao, S.; Grishman, R. Extracting Relations with Integrated Information Using Kernel Methods. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Harbor, MI, USA, 25–30 June 2005; pp. 419–426.
8. Zhou, G.; Su, J.; Zhang, J.; Zhang, M. Exploring Various Knowledge in Relation Extraction. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL), Ann Harbor, MI, USA, 25–30 June 2005; pp. 427–434.
9. Bach, N.; Badaskar, S. A Review of Relation Extraction. Available online: orb.essex.ac.uk/CE/CE807/Readings/A-survey-on-Relation-Extraction.pdf (accessed on 20 May 2016).
10. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant Supervision for Relation Extraction without Labeled Data. In Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1003–1011.
11. Riedel, S.; Yao, L.; McCallum, A. Modeling Relations and Their Mentions without Labeled Text. In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain, 20–24 September 2010; pp. 148–163.
12. Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D.S. Knowledge Based Weak Supervision for Information Extraction of Overlapping Relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), Portland, OR, USA, 19–24 June 2011; pp. 541–550.
13. Surdeanu, M.; Tibshirani, J.; Nallapati, R.; Manning, C.D. 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 (EMNLP), Jeju Island, Korea, 12–14 July 2012; pp. 455–465.
14. Min, B.; Grishman, R.; Wan, L.; Wang, C.; Gondek, D. Distant Supervision for Relation Extraction with an Incomplete Knowledge Base. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Atlanta, GA, USA, 9–15 June 2013; pp. 777–782.
15. Ritter, A.; Zettlemoyer, L.; Etzioni, O. Modeling Missing Data in Distant Supervision for Information Extraction. *Trans. Assoc. Comput. Linguist.* **2013**, *1*, 367–378.
16. Craven, M.; Kumlien, J. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB), Heidelberg, Germany, 6–10 August 1999; pp. 77–86.
17. Bunescu, R.C.; Mooney, R.J. Learning to Extract Relations from the Web Using Minimal Supervision. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech, 23–30 June 2007; pp. 576–583.
18. Bellare, K.; McCallum, A. Learning Extractors from Unlabeled Text Using Relevant Databases. Available online: http://www.aaai.org/Papers/Workshops/2007/WS-07-14/WS07-14-002.pdf (accessed on 20 May 2016).
19. Wu, F.; Weld, D. Autonomously Semantifying Wikipedia. In Proceedings of the 16th International Conference on Information and Knowledge Management (CIKM), Lisbon, Portugal, 6–10 November 2007; pp. 41–50.
20. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the 26th Advances in Neural Information Processing Systems (NIPS), South Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
21. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
22. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
23. Fan, M.; Zhao, D.; Zhou, Q.; Liu, Z.; Zheng, T.F.; Chang, E.Y. Distant Supervision for Relation Extraction with Matrix Completion. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Baltimore, MD, USA, 22–27 June 2014; pp. 839–849.
24. Angeli, G.; Tibshirani, J.; Wu, J.Y.; Manning, C.D. Combining Distant and Partial Supervision for Relation Extraction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1556–1567.

25. Nagesh, A.; Haffari, G.; Ramakrishna, G. Noisy-or Based Model for Relation Extraction Using Distant Supervision. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1937–1941.
26. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 1753–1762.
27. Intxaurrondo, A.; Surdeanu, M.; de Lacalle, O.L.; Agirre, E. Removing Noisy Mentions for Distant Supervision. *Proces. Leng. Nat.* **2013**, *51*, 41–48.
28. Xu, W.; Hoffmann, R.; Zhao, L.; Grishman, R. Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), Sofia, Bulgaria, 4–9 August 2013; pp. 665–670.
29. Takamatsu, S.; Sato, I.; Nakagawa, H. Reducing Wrong Labels in Distant Supervision for Relation Extraction. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), Jeju Island, Korea, 8–14 July 2012; pp. 721–729.
30. Xiang, Y.; Zhang, Y.; Wang, X.; Qin, Y.; Han, W. Bias Modeling for Distantly Supervised Relation Extraction. *Math. Probl. Eng.* **2015**, *2015*, 969053. [CrossRef]
31. Xiang, Y.; Wang, X.; Zhang, Y.; Qin, Y.; Fan, S. Distant Supervision for Relation Extraction via Group Selection. In Proceedings of the 22nd International Conference on Neural Information Processing (ICONIP), Istanbul, Turkey, 9–12 November 2015; pp. 250–258.
32. Usunier, N.; Buffoni, D.; Gallinari, P. Ranking with Ordered Weighted Pairwise Classification. In Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, QC, Canada, 14–18 June 2009; pp. 1057–1064.
33. Weston, J.; Bengio, S.; Usunier, N. Wsabie: Scaling up to Large Vocabulary Image Annotation. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain, 16–22 July 2011; pp. 2764–2770.
34. Huang, S.; Gao, W.; Zhou, Z.H. Fast Multi-Instance Multi-Label Learning. In Proceedings of the 2014 AAAI Conference on Artificial Intelligence (AAAI), Hilton, QC, Canada, 27–31 July 2014; pp. 1868–1874.
35. Ji, H.; Grishman, R.; Dang, H.T.; Griffitt, K.; Ellis, J. Overview of the TAC 2010 Knowledge Base Population Track. In Proceedings of the Third Text Analysis Conference (TAC 2010), Gaithersburg, MD, USA, 15–16 November 2010.
36. Ji, H.; Grishman, R.; Dang, H.T. Overview of the TAC 2011 Knowledge Base Population Track. In Proceedings of the Forth Text Analytics Conference (TAC 2011), Gaithersburg, MD, USA, 14–15 November 2011.