Encoding Relation Requirements for Relation Extraction via Joint Inference

Liwei Chen[1], **Yansong Feng***[1], **Songfang Huang**[2], **Yong Qin**[2] and **Dongyan Zhao**[1]
[1]ICST, Peking University, Beijing, China
[2]IBM China Research Lab, Beijing, China
chenliwei,fengyansong,zhaodongyan@pku.edu.cn
huangsf,qinyong@cn.ibm.com

Abstract

Most existing relation extraction models make predictions for each entity pair locally and individually, while ignoring implicit global clues available in the knowledge base, sometimes leading to conflicts among local predictions from different entity pairs. In this paper, we propose a joint inference framework that utilizes these global clues to resolve disagreements among local predictions. We exploit two kinds of clues to generate constraints which can capture the implicit type and cardinality requirements of a relation. Experimental results on three datasets, in both English and Chinese, show that our framework outperforms the state-of-the-art relation extraction models when such clues are applicable to the datasets. And, we find that the clues learnt automatically from existing knowledge bases perform comparably to those refined by human.

1 Introduction

Identifying predefined kinds of relationship between pairs of entities is crucial for many knowledge base related applications(Suchanek et al., 2013). In the literature, relation extraction (RE) is usually investigated in a classification style, where relations are simply treated as isolated class labels, while their definitions or background information are sometimes ignored. Take the relation *Capital* as an example, we can imagine that this relation will expect a country as its subject and a city as object, and in most cases, a city can be the capital of only one country. All these clues are no doubt helpful, for instance, Yao et al. (2010) explicitly modeled the expected types of a relation's arguments with the help of Freebase's type taxonomy and obtained promising results for RE.

*Yansong Feng is the corresponding author.

However, properly capturing and utilizing such typing clues are not trivial. One of the hurdles here is the lack of off-the-shelf resources and such clues often have to be coded by human experts. Many knowledge bases do not have a well-defined typing system, let alone fine-grained typing taxonomies with corresponding type recognizers, which are crucial to explicitly model the typing requirements for arguments of a relation, but rather expensive and time-consuming to collect. Similarly, the cardinality requirements of arguments, e.g., a person can have only one birthdate and a city can only be labeled as capital of one country, should be considered as a strong indicator to eliminate wrong predictions, but has to be coded manually as well.

On the other hand, most previous relation extractors process each entity pair (we will use *entity pair* and *entity tuple* exchangeably in the rest of the paper) locally and individually, i.e., the extractor makes decisions solely based on the sentences containing the current entity pair and ignores other related pairs, therefore has difficulties to capture possible disagreements among different entity pairs. However, when looking at the output of a multi-class relation predictor globally, we can easily find possible incorrect predictions such as a university locates in two different cities, two different cities have been labeled as capital for one country, a country locates in a city and so on.

In this paper, we will address how to derive and exploit two categories of these clues: the expected types and the cardinality requirements of a relation's arguments, in the scenario of relation extraction. We propose to perform joint inference upon multiple local predictions by leveraging implicit clues that are encoded with relation specific requirements and can be learnt from existing knowledge bases. Specifically, the joint inference framework operates on the output of a sentence level relation extractor as input, derives 5 types of constraints from an existing KB to implicitly capture

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 818–827,
Baltimore, Maryland, USA, June 23-25 2014.

the expected type and cardinality requirements for a relation's arguments, and jointly resolve the disagreements among candidate predictions. We formalize this procedure as a constrained optimization problem, which can be solved by many optimization frameworks. We use integer linear programming (ILP) as the solver and evaluate our framework on English and Chinese datasets. The experimental results show that our framework performs better than the state-of-the-art approaches when such clues are applicable to the datasets. We also show that the automatically learnt clues perform comparably to those refined manually.

In the rest of the paper, we first review related work in Section 2, and in Section 3, we describe our framework in detail. Experimental setup and results are discussed in Section 4. We conclude this paper in Section 5.

2 Related Work

Since traditional supervised relation extraction methods (Soderland et al., 1995; Zhao and Grishman, 2005) require manual annotations and are often domain-specific, nowadays many efforts focus on semi-supervised or unsupervised methods (Banko et al., 2007; Fader et al., 2011). Distant supervision (DS) is a semi-supervised RE framework and has attracted many attentions (Bunescu, 2007; Mintz et al., 2009; Yao et al., 2010; Surdeanu et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012). DS approaches can predict canonicalized (predefined in KBs) relations for large amount of data and do not need much human involvement. Since the automatically generated training datasets in DS often contain noises, there are also research efforts focusing on reducing the noisy labels in the training data (Takamatsu et al., 2012). To bridge the gaps between the relations extracted from open information extraction and the canonicalized relations in KBs, Yao et al. (2012) and Riedel et al. (2013) propose a universal schema which is a union of KB schemas and natural language patterns, making it possible to integrate the unlimited set of uncanonicalized relations in open settings with the relations in existing KBs.

As far as we know, few works have managed to take the relation specific requirements for arguments into account, and most existing works make predictions locally and individually. The MultiR system allows entity tuples to have more than one relations, but still predicts each entity tuple locally (Hoffmann et al., 2011). Surdeanu et al. (2012) propose a two-layer multi-instance multi-label (MIML) framework to capture the dependencies among relations. The first layer is a multi-class classifier making local predictions for single sentences, the output of which are aggregated by the second layer into the entity pair level. Their approach only captures relation dependencies, while we learn implicit relation backgrounds from knowledge bases, including argument type and cardinality requirements. Riedel et al. (2013) propose to use latent vectors to estimate the preferences between relations and entities. These can be considered as the latent type information of the relations' arguments, which is learnt from various data sources. In contrast, our approach learn implicit clues from existing KBs, and jointly optimize local predictions among different entity tuples to capture both relation argument type clues and cardinality clues. Li et al. (2011) and Li et al. (2013) use co-occurring statistics among relations or events to jointly improve information extraction performances in ACE tasks, while we mine existing KBs to collect global clues to solve local conflicts and find the optimal aggregation assignments, regarding existing knowledge facts. de Lacalle and Lapata (2013) encode general domain knowledge as FOL rules in a topic model while our instantiated constraints are directly operated in an ILP model. Zhang et al. (2013) utilize relation cardinality to create negative samples for distant supervision while we use both implicit type clues and relation cardinality expectations to discover possible inconsistencies among local predictions.

3 The Framework

Our framework takes a set of entity pairs and their supporting sentences as its input. We first train a preliminary sentence level extractor which can output confidence scores for its predictions, e.g., a maximum entropy or logistic regression model, and use this local extractor to produce local predictions. In order to implicitly capture the expected type and cardinality requirements for a relation's arguments, we derive two kinds of clues from an existing KB, which are further utilized to discover the disagreements among local candidate predictions. Our objective is to maximize the overall confidence of all the selected predictions.

3.1 Generating Candidate Relations

Since we will focus on the open domain relation extraction, we still follow the distant supervision paradigm to collect our training data guided by a KB, and train the local extractor accordingly. Specifically, we train a sentence level extractor using the maximum entropy model. Given a sentence containing an entity pair, the model will output the confidence of this sentence representing certain relationship (from a predefined relation set) between the entity pair. Formally $\mathcal{R}$ represents the relation set we are working on, $\mathcal{T}$ is the set of entity tuples that we will predict in the test set.

Keep in mind that our local extractor is trained on noisy training data, which, we admit, is not fully reliable. As we observed in a pilot experiment that there is a good chance that the predictions ranked in the second or third may still be correct, we select **top three** predictions as the candidate relations for each mention in order to introduce more potentially correct output.

On the other hand, we should discard the predictions whose confidences are too low to be true, where we set up a threshold of 0.1. For a tuple t, we obtain its candidate relation set by combining the candidate relations of all its mentions, and represent it as R^t. For a candidate relation $r \in R^t$ and a tuple t, we define M_t^r as all t's mentions whose candidate relations contain r. Now the confidence score of a relation $r \in R^t$ being assigned to tuple t can be calculated as:

$$conf(t, r) = \sum_{m \in M_t^r} \text{MEscore}(m, r) \qquad (1)$$

where MEscore(m, r) is the confidence of mention m representing relation r output by our preliminary extractor.

Traditionally, both lexical features and syntactic features are used in relation extraction. Lexical features are the word chains between the subjects and objects in the sentences, while syntactic features are the dependency paths from the subjects to the objects on the dependency graphs of the supporting sentences. However, lexical features are usually too specific to frequently appear in the test data, while the reliability of syntactic features depends heavily on the quality of dependency parsing tools. Generally, we expect more potentially correct relations to be put into the candidate relation set for further consideration. So in addition to lexical and syntactic features, we also use n-gram features to train our preliminary relation extraction model. N-gram features are considered as more ambiguous compared to traditional lexical and syntactic features, and may introduce incorrect predictions, thus improving the recall at the cost of precision.

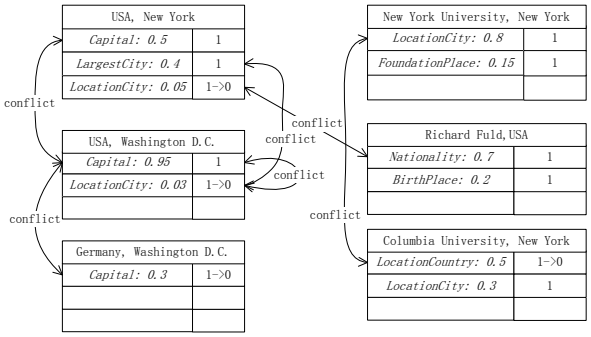


Figure 1: The different types of disagreements we will investigate in the candidate relations. The clues of detecting these inconsistencies can be learnt from a knowledge base.

3.2 Disagreements among the Candidates

The candidate relations we obtained in the previous subsection inevitably include many incorrect predictions. Ideally we should discard those wrong predictions to produce more accurate results.

As discussed earlier, we will exploit from the knowledge base two categories of clues that implicitly capture relations' backgrounds: their expected argument types and argument cardinalities, based on which we can discover two categories of disagreements among the candidate predictions, summarized as argument type inconsistencies and violations of arguments' uniqueness, which have been rarely considered before. We will discuss them in detail, and describe how to learn the clues from a KB afterwards.

Implicit Argument Types Inconsistencies: Generally, the argument types of the correct predictions should be consistent with each other. Given a relation, its arguments sometimes are required to be certain types of entities. For example, in Figure 1, the relation *LargestCity* restricts its subject to be either countries or states, and its object to be cities. If the predictions among different entity tuples require the same entity to belong to different types, we call this

an argument type inconsistency. Take <USA, New York> and <USA, Washington D.C.> as an example. In Figure 1, <USA, New York> has a candidate relation *LargestCity* which restricts *USA* to be either countries or states, while <USA, Washington D.C.> has a prediction *LocationCity* which indicates a disagreement in terms of *USA*'s type because the latter prediction expects *USA* to be an organization located in a city. This warns that at least one of the two candidate relations is incorrect.

The previous scenario shows that the subjects of two candidate relations may disagree with each other. From Figure 1, we can observe two more situations: the first one is that the objects of the two candidate relations are inconsistent with each other, for example <New York University, New York> with the prediction *LocationCity* and <Columbia University, New York> with the prediction *LocationCountry*. The second one is that the subject of one candidate relation do not agree with another prediction's object, for example <Richard Fuld, USA> with the prediction *Nationality* and <USA, New York> with the prediction *LocationCity*. Although we have not assigned explicit types to these entities, we can still exploit the inconsistencies implicitly with the help of shared entities. Note that the implicit argument typing clues here mean whether two relations can share arguments, but NOT enumate what types explicitly their arguments should have.

We formalize all the relation pairs that disagree with each other as follows. These relation pairs can be divided into three subcategories. We represent the relation pairs (r_i, r_j) that are inconsistent in terms of subjects as $\mathcal{C}^{sr}$, the relations pairs that are inconsistent in terms of objects as $\mathcal{C}^{ro}$, the relation pairs that are inconsistent in terms of one's subject and the other one's object as $\mathcal{C}^{rer}$.

It is worth mentioning that disagreements inside a tuple are also included here. For instance, an entity tuple <USA, Washington D.C.> in Figure 1 has two candidate relations, *Capital* and *LocationCity*. These two predictions are inconsistent with each other with respect to the type of *USA*. They implicitly consider *USA* as "country" and "organization", respectively.

Violations of Arguments' Uniqueness: The previous categories of disagreements are all based on the implicit type information of the relations' arguments, Now we make use of the clues of argument cardinality requirements. Given a subject, some relations should have unique objects. For example, in Figure 1, given *USA* as the subject of the relation *Capital*, we can only accept one possible object, because there is great chance that a country only have one capital. On the other hand, given *Washington D.C.* as the object of the relation *Capital*, we can only accept one subject, since usually a city can only be the capital of one country or state. If these are violating in the candidates, we could know that there may be some incorrect predictions. We represent the relations expecting unique objects as $\mathcal{C}^{ou}$, and the relations expecting unique subjects as $\mathcal{C}^{su}$.

3.3 Obtaining the Global Clues

Now, the issue is how to obtain the clues used in the previous subsection. That is, how we determine which relations expect certain types of subjects, which relations expect certain types of objects, etc. These knowledge can be definitely coded by human, or learnt from a KB.

Most existing knowledge bases represent their knowledge facts in the form of (<*subject, relation, object*>) triple, which can be seen as relational facts between entity tuples. Usually the triples in a KB are carefully defined by experts. It is rare to find inconsistencies among the triples in the knowledge base. The clues are therefore learnt from KBs, and further refined manually if needed.

Given two relations r_1 and r_2, we query the KB for all tuples bearing the relation r_1 or r_2. We use S_i and O_i to represent r_i's ($i \in \{1, 2\}$) subject set and object set, respectively. We adopt the pointwise mutual information (PMI) to estimate the dependency between the argument sets of two relations:

$$\text{PMI}(A, B) = \log \frac{p(A, B)}{p(A)p(B)} \qquad (2)$$

where $p(A, B)$ is number of the entities both in A and B, $p(A)$ and $p(B)$ are the numbers of the entities in A and B, respectively. For any pair of relations from $\mathcal{R} \times \mathcal{R}$, we calculate four scores: PMI(S_1, S_2), PMI(O_1, O_2), PMI(S_1, O_2) and PMI(S_2, O_1). To make more stable estimations, we set up a threshold for the PMI. If PMI(S_1, S_2) is lower than the threshold, we will consider that r_1 and r_2 cannot share a subject. Things are similar for the other three scores. The threshold is set to -3 in this paper.

We can also learn the uniqueness of arguments for relations. For each pre-defined relation in $\mathcal{R}$, we collect all the triples containing this relation, and count the portion of the triples which only have one object for each subject, and the portion of the triples which only have one subject for each object. The relations whose portions are higher than the threshold will be considered to have unique argument values. This threshold is set to 0.8 in this paper.

3.4 Integer Linear Program Formulation

As discussed above, given a set of entity pairs and their candidate relations output by a preliminary extractor, our goal is to find an optimal configuration for all those entities pairs jointly, solving the disagreements among those candidate predictions and maximizing the overall confidence of the selected predictions. This is an NP-hard optimization problem. Many optimization models can be used to obtain the approximate solutions.

In this paper, we propose to solve the problem by using an ILP tool, IBM ILOG Cplex[1]. Firstly, for each tuple t and one of its candidate relations r, we define a binary decision variable d_t^r indicating whether the candidate relation r is selected by the solver. Our objective is to maximize the total confidence of all the selected candidates, and the objective function can be written as:

$$\max \sum_{t\in\mathcal{T}, r\in R^t} conf(t,r) d_t^r + \sum_{\forall t, r\in R^t, m\in M_t^r} \max \text{MEscore}(m,r) d_t^r$$

where $conf(t,r)$ is the confidence of the tuple t bearing the candidate relation r. The first component is the sum of the original confidence scores of all the selected candidates, and the second one is the sum of the maximal mention-level confidence scores of all the selected candidates. The latter is designed to encourage the model to select the candidates with higher individual mention-level confidence scores.

We add the constraints with respect to the disagreements described in Section 3.2. For the sake of clarity, we describe the constraints derived from each scenario of the two categories of disagreements separately.

The subject-relation constraints avoid the disagreements between the predictions of two tuples sharing a subject. These constraints can be represented as:

$$d_{t_i}^{r^{t_i}} + d_{t_j}^{r^{t_j}} \leq 1 \quad (3)$$
$$\forall t_i, t_j : subj(t_i) = subj(t_j) \wedge (r^{t_i}, r^{t_j}) \in \mathcal{C}^{sr}$$

where t_i and t_j are two tuples in $\mathcal{T}$, $subj(t_i)$ is the subject of t_i, r^{t_i} is a candidate relation of t_i, r^{t_j} is a candidate relation of t_j.

The object-relation constraints avoid the inconsistencies between the predictions of two tuples sharing an object. Formally we add the following constraints:

$$d_{t_i}^{r^{t_i}} + d_{t_j}^{r^{t_j}} \leq 1 \quad (4)$$
$$\forall t_i, t_j : obj(t_i) = obj(t_j) \wedge (r^{t_i}, r^{t_j}) \in \mathcal{C}^{ro}$$

where $t_i \in \mathcal{T}$ and $t_j \in \mathcal{T}$ are two tuples, $obj(t_i)$ is the object of t_i.

The relation-entity-relation constraints ensure that if an entity works as subject and object in two tuples t_i and t_j respectively, their relations agree with each other. The constraints we add are:

$$d_{t_i}^{r^{t_i}} + d_{t_j}^{r^{t_j}} \leq 1 \quad (5)$$
$$\forall t_i, t_j : obj(t_i) = subj(t_j) \wedge (r^{t_i}, r^{t_j}) \in \mathcal{C}^{rer}$$

The object uniqueness constraints ensure that the relations requiring unique objects do not bear more than one object given a subject.

$$\sum_{t\in Tuple(r), subj(t)=e} d_t^r \leq 1 \quad (6)$$
$$\forall e \wedge r \in \mathcal{C}^{ou}$$

where e is an entity, $Tuple(r)$ are the tuples whose candidate relations contain r.

The subject uniqueness constraints ensure that given an object, the relations expecting unique subjects do not bear more than one subject.

$$\sum_{t\in Tuple(r), obj(t)=e} d_t^r \leq 1 \quad (7)$$
$$\forall e \wedge r \in \mathcal{C}^{su}$$

By adopting ILP, we can combine the local information including MaxEnt confidence scores and the implicit relation backgrounds that are embedded into global consistencies of the entity tuples together. After the optimization problem is solved, we will obtain a list of selected candidate relations for each tuple, which will be our final output.

[1] www.cplex.com

4 Experiments

4.1 Datasets

We evaluate our approach on three datasets, including two English datasets and one Chinese dataset.

The first English dataset, Riedel's dataset, is the one used in (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012), with the same split. It uses Freebase as the knowledge base and New York Time corpus as the text corpus, including about 60,000 entity tuples in the training set, and about 90,000 entity tuples in the testing set.

We generate the second English dataset, DBpedia dataset, by mapping the triples in DBpedia (Bizer et al., 2009) to the sentences in New York Time corpus. We map 51 different relations to the corpus and result in about 50,000 entity tuples, 134,000 sentences for training and 30,000 entity tuples, 53,000 sentences for testing.

For the Chinese dataset, we derive knowledge facts and construct a Chinese KB from the Infoboxes of HudongBaike, one of the largest Chinese online encyclopedias. We collect four national economic newspapers in 2009 as our corpus. 28 different relations are mapped to the corpus and this results in 60,000 entity tuples, 120,000 sentences for training and 40,000 tuples, 83,000 sentences for testing.

4.2 Baselines and Competitors

The baseline we use in this paper is Mintz++, which is described in (Surdeanu et al., 2012). It is a modification of the model proposed by Mintz et al. (2009). The model predicts for each mention separately, and allows multi-label outputs for an entity tuple by OR-ing the outputs of its mentions.

As we described in Section 3.1, originally we select the top three predicted relations as the candidates for each mention. In order to investigate whether it is necessary to use up to three candidates, we implement two variants of our approach, which select the top one and top two relations as candidates for each mention, and represented as ILP-1cand and ILP-2cand, respectively.

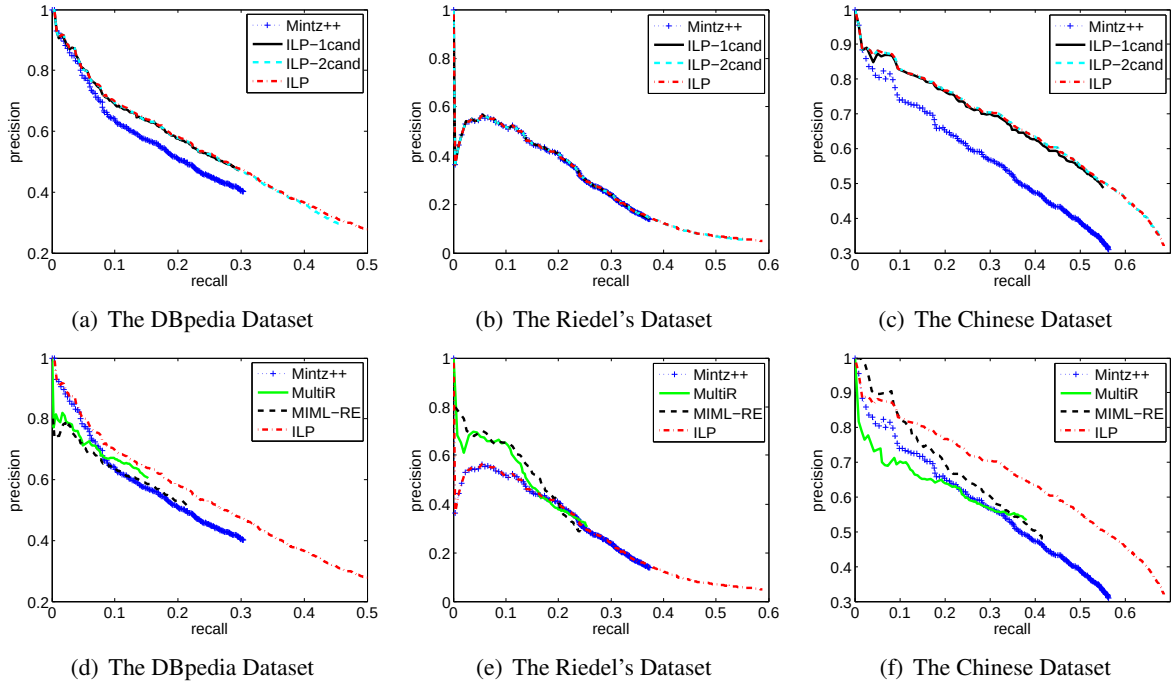
We also use two distant supervision approaches for the comparison. The first one is MultiR (Hoffmann et al., 2011), a novel joint model that can deal with the relation overlap issue. The second one, MIML-RE (Surdeanu et al., 2012), is one of the state-of-the-art MIML relation extraction systems. We tune the models of MultiR and MIML-RE so that they fit our datasets.

4.3 Overall Performance

First we compare our framework and its variants with the baseline and the state-of-the-art RE models. Following previous works, we use the Precision-Recall curve as the evaluation criterion in our experiment. The results are summarized in Figure 2. For the constraints, we first manually select an average of 20 relation pairs for each subcategory of the first kind of clues, and all the relations with unique argument values in $\mathcal{R}$. We also show how automatically learnt clues perform in Section 4.5.

Figure 2 shows that compared with the baseline, our framework performs consistently better in the DBpedia dataset and the Chinese dataset. Mintz++ proves to be a strong baseline on both datasets. It tends to result in a high recall, and its weakness of low precision is perfectly fixed by the ILP model. Our ILP model and its variants all outperform Mintz++ in precision in both datasets, indicating that our approach helps filter out incorrect predictions from the output of MaxEnt model. Compared with MultiR, our framework obtains better results in both datasets. Especially in the Chinese dataset, the improvement in precision reaches as high as 10-16% at the same recall points. Our framework performs better compared to MIML-RE in the English dataset. On the Chinese dataset, our framework outperforms MIML-RE except in the low-recall portion ($<$10%) of the P-R curve. All these results show that embedding the relation background information into RE can help eliminate the wrong predictions and improve the results.

However, in the Riedel's dataset, Mintz++, the MaxEnt relation extractor, does not perform well, and our framework cannot improve its performance. In order to find out the reasons, we manually investigate the dataset. The top three relations of this dataset are */location/location/contains*, */people/person/nationality* and */people/person/place_lived*. About two-thirds of the entity tuples belongs to these three relations, and the outputs of the local extractor usually bias even more to the large relations. What is worse, we cannot find any clues from the top three relations because their arguments' types are too general. Things are similar for many other



(a) The DBpedia Dataset (b) The Riedel's Dataset (c) The Chinese Dataset

(d) The DBpedia Dataset (e) The Riedel's Dataset (f) The Chinese Dataset

Figure 2: Overall performances of our framework and its variants, the baselines and the state-of-the-art approaches on the three datasets.

relations in this dataset. Although we may find some clues any way, they are too few to make any improvement. Hence, our framework does not perform well due to the poor performance of MaxEnt extractor and the lack of clues. To solve this problem, we think of addressing the selection preferences between relations and entities proposed in (Riedel et al., 2013), which should be our future work.

We notice that in all three datasets our variant ILP-1cand is shorter than Mintz++ in recall, indicating we may incorrectly discard some predictions. Compared to ILP-2cand and original ILP, ILP-1cand leads to slightly lower precision but much lower recall, showing that selecting more candidates may help us collect more potentially correct predictions. Comparing ILP-2cand and original ILP, the latter hardly makes any improvement in precision, but is slightly longer in recall, indicating using three candidates can still collect some more potentially correct predictions, although the number may be limited.

In order to study how our framework improves the performances on the DBpedia dataset and the Chinese dataset, we further investigate the number of incorrect predictions eliminated by ILP and the number of incorrect predictions corrected by ILP. We also examine the number of correct predictions newly introduce by ILP, which were NA in Mintz++. We summarize the results in Table 1.

Table 1: Details of the improvements made by ILP in the DBpedia and Chinese datasets.

Datasets	Incorrect Predictions Eliminated	Wrong Predictioins Corrected	Correct Predictions Newly Introduced
DBpedia	268	61	1426
Chinese	1506	14	283

The results show that our framework can reduce the incorrect predictions and introduce more correct predictions at the same time. We also find an interesting results: in the DBpedia dataset, ILP is more likely to introduce correct predictions to the results, while in the Chinese dataset it tends to reduce more incorrect predictions, which may be caused by the differences between performances of Mintz++ on the two datasets, where it gets a higher recall on the Chinese dataset.

Following Surdeanu et al. (2012), we also list the peak F1 score (highest F1 score) for each model in Table 2. Different from (Surdeanu et al., 2012), we use all the entity pairs instead of the ones with more than 10 mentions. We can observe that our model obtains the best performance in the DBpedia dataset and the Chinese dataset. In the DBpedia dataset, it is 3.6% higher than Mintz++,

7.9% higher than MIML-RE and 13.9% higher than MultiR. In the Chinese dataset, Mintz++, MultiR and MIML-RE performs similarly in terms of the highest F1 score, while our model gains about 8% improvement. In the Riedel's dataset, our framework hardly obtains any improvement compared with Mintz++.

We also investigate the impacts of the constraints used in ILP, which are derived based on the two kinds of clues and can encode relation definition information into our framework. Experimental results in Table 2 shows that in the DBpedia dataset, the highest F1 score increases from 35.2% to 38.3% with the help of both kinds of clues, while in the Chinese dataset the improvement is from 44.4% to 52.8%. In the Riedel's dataset we do not see any improvements since there are almost no clues. Furthermore, using constraints derived from only one kind of clues can also improve the performance, but not as well as using both of them.

4.4 Adapting MultiR Sentence Level Extractor to Our Framework

The preliminary relation extractor of our optimization framework is not limited to the MaxEnt extractor, and can take any sentence level relation extractor with confidence scores. We also fit MultiR's mention level extractor into our framework.

As shown in Figure 3, in the DBpedia dataset and the Chinese dataset, in most parts of the curve, ILP optimized MultiR outperforms original MultiR. We think the reason is that our framework make use of global clues to discard the incorrect predictions. The results are not as high as when we use MaxEnt as the preliminary extractor. We think one reason is that MultiR does not perform well in these two datasets. Furthermore, the confidence scores which MultiR outputs are not normalized to the same scale, which brings us difficulties in setting up a confidence threshold to select the candidates. As a result, we only use the top one result as the candidate since including top two predictions without thresholding the confidences performs bad, indicating that a probabilistic sentence-level extractor is more suitable for our framework. We also notice that in the Riedel's dataset our framework does not improve the performance significantly, and we have discussed the reasons in Section 4.3.

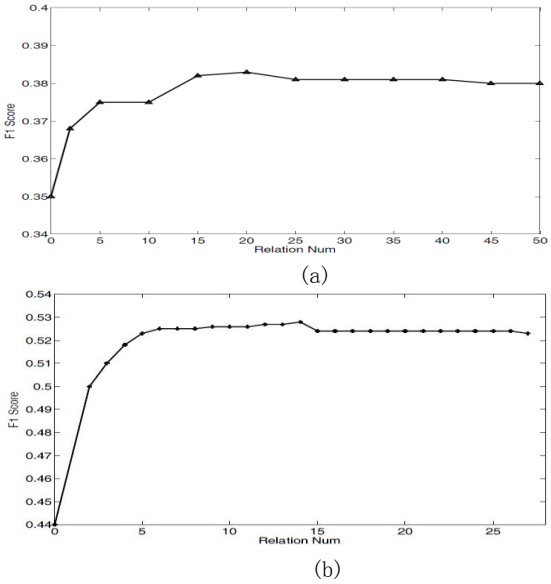
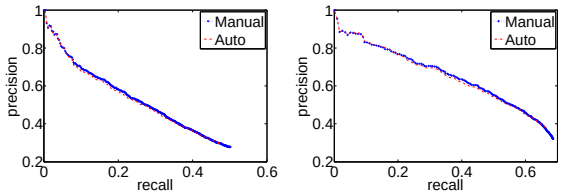


Figure 4: F1 score v.s. number of relations (used to introduce the related learnt clues into the ILP framework) on the DBpedia dataset (a) and the Chinese dataset (b).



(a) The DBpedia Dataset (b) The Chinese Dataset

Figure 5: Performances of manually selected clues and automatically learnt clues on two datasets.

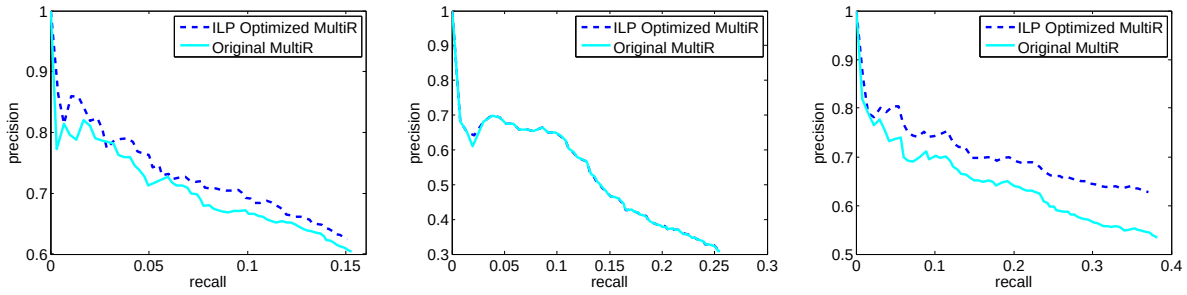
4.5 Examining the Automatically Learnt Clues

Now we evaluate the performance of automatically collected clues used in our model. Since there are almost no clues in the Riedel's dataset, we only investigate the other two datasets. We add clues according to their related relations' proportions in the local predictions. For example, *Country* and *birthPlace* take up about 30% in the local predictions, we thus add clues that are related to these two relations, and then move on with new clues related to other relations according to those relations' proportions in the local predictions.

As is shown in Figure 4, in both datasets, the clues related to more local predictions will solve more inconsistencies, thus are more effective. Adding the first two relations improves the model significantly, and as more relations are added, the

Table 2: Results of the highest F1 score on all three datasets.

	DBpedia			Riedel			Chinese		
Method	**P(%)**	**R(%)**	**F1(%)**	**P(%)**	**R(%)**	**F1(%)**	**P(%)**	**R(%)**	**F1(%)**
Mintz++	40.2	30.5	34.7	35.3	23.2	27.9	43.3	45.7	44.4
MultiR	60.4	15.3	24.4	32.3	25.1	**28.2**	53.5	38.2	44.6
MIML-RE	51.3	21.6	30.4	41.5	19.9	26.9	49.2	41.3	44.9
ILP	37.4	39.2	**38.3**	35.5	23.2	28.0	52.6	52.9	**52.8**
ILP-No-Constraint	34.1	36.3	35.2	35.3	23.2	28.0	43.3	45.7	44.4
ILP-Type-Inconsistent	36.3	39.2	37.7	35.5	23.2	28.0	49.5	49.0	49.2
ILP-Cardinality	35.3	37.8	36.5	35.4	23.2	28.0	50.3	48.8	49.6



(a) The DBpedia Dataset (b) The Riedel's Dataset (c) The Chinese Dataset

Figure 3: The results of original MultiR and ILP optimized MultiR on the three datasets.

performances keep increasing until approaching the still state. It is worth mentioning that when sufficient learnt clues are added into the model, the results are comparable to those based on the clues refined manually, as shown in Figure 5. This indicates that the clues can be collected automatically, and further used to examine whether predicted relations are consistent with the existing ones in the KB, which can be considered as a form of quality control.

5 Conclusions

In this paper, we make use of the global clues derived from KB to help resolve the disagreements among local relation predictions, thus reduce the incorrect predictions and improve the performance of relation extraction. Two kinds of clues, including implicit argument type information and argument cardinality information of relations are investigated. Our framework outperforms the state-of-the-art models if we can find such clues in the KB. Furthermore, our framework is scalable for other local sentence level extractors in addition to the MaxEnt model. Finally, we show that the clues can be learnt automatically from the KB, and lead to comparable performance to manually refined ones.

For future work, we will investigate other kinds of clues and attempt a joint optimization framework that could host entity disambiguation, relation extraction and entity linking together. We will also adopt selection preference between entities and relations since sometimes we may not find useful clues.

Acknowledgments

We would like to thank Heng Ji, Dong Wang and Kun Xu for their useful discussions and the anonymous reviewers for their helpful comments which greatly improved the work. This work was supported by the National High Technology R&D Program of China (Grant No. 2012AA011101), National Natural Science Foundation of China (Grant No. 61272344, 61202233, 61370055) and the joint project with IBM Research.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*, IJCAI'07, pages 2670–2676.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak,

and Sebastian Hellmann. 2009. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7:154–165, September.

Razvan C. Bunescu. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL07*.

Oier Lopez de Lacalle and Mirella Lapata. 2013. Unsupervised relation extraction with general domain knowledge. In *EMNLP*, pages 415–425. ACL.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th ACL-HLT - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA. ACL.

Qi Li, Sam Anzaroot, Wen-Pin Lin, Xiang Li, and Heng Ji. 2011. Joint inference for cross-document information extraction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2225–2228, New York, NY, USA. ACM.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*, pages 73–82. The Association for Computer Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 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 IJCNLP of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer Berlin / Heidelberg.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, June.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal inducing a conceptual dictionary. In *Proceedings of the 14th IJCAI - Volume 2*, IJCAI'95, pages 1314–1319, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Fabian Suchanek, James Fan, Raphael Hoffmann, Sebastian Riedel, and Partha Pratim Talukdar. 2013. Advances in automated knowledge base construction. In *SIGMOD Records journal*, March.

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. 2010. A simple distant supervision approach for the TAC-KBP slot filling task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA, November.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*, pages 455–465. ACL.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 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*, ACL '12, pages 721–729, Stroudsburg, PA, USA. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of EMNLP*, EMNLP '10, pages 1013–1023, Stroudsburg, PA, USA. ACL.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Probabilistic databases of universal schema. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 116–121, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xingxing Zhang, Jianwen Zhang, Junyu Zeng, Jun Yan, Zheng Chen, and Zhifang Sui. 2013. Towards accurate distant supervision for relational facts extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 810–815, Sofia, Bulgaria, August. Association for Computational Linguistics.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 419–426, Stroudsburg, PA, USA. Association for Computational Linguistics.