Towards Accurate Distant Supervision for Relational Facts Extraction

Xingxing Zhang[1] Jianwen Zhang[2]* Junyu Zeng[3] Jun Yan[2] Zheng Chen[2] Zhifang Sui[1]
[1]Key Laboratory of Computational Linguistics (Peking University), Ministry of Education,China
[2]Microsoft Research Asia
[3]Beijing University of Posts and Telecommunications
[1]{zhangxingxing,szf}@pku.edu.cn
[2]{jiazhan,junyan,zhengc}@microsoft.com
[3]junyu.zeng@gmail.com

Abstract

Distant supervision (DS) is an appealing learning method which learns from existing relational facts to extract more from a text corpus. However, the accuracy is still not satisfying. In this paper, we point out and analyze some critical factors in DS which have great impact on accuracy, including valid entity type detection, negative training examples construction and ensembles. We propose an approach to handle these factors. By experimenting on Wikipedia articles to extract the facts in Freebase (the top 92 relations), we show the impact of these three factors on the accuracy of DS and the remarkable improvement led by the proposed approach.

1 Introduction

Recently there are great efforts on building large structural knowledge bases (KB) such as Freebase, Yago, etc. They are composed of relational facts often represented in the form of a triplet, `(SrcEntity, Relation, DstEntity)`, such as "(Bill Gates, BornIn, Seattle)". An important task is to enrich such KBs by extracting more facts from text. Specifically, this paper focuses on extracting facts for existing relations. This is different from OpenIE (Banko et al., 2007; Carlson et al., 2010) which needs to discover new relations.

Given large amounts of labeled sentences, supervised methods are able to achieve good performance (Zhao and Grishman, 2005; Bunescu and Mooney, 2005). However, it is difficult to handle large scale corpus due to the high cost of labeling. Recently an approach called distant supervision (DS) (Mintz et al., 2009) was proposed, which does not require any labels on the text. It treats the extraction problem as classifying a candidate entity pair to a relation. Then an existing fact in a KB can be used as a labeled example whose label is the relation name. Then the features of all the sentences (from a given text corpus) containing the entity pair are merged as the feature of the example. Finally a multi-class classifier is trained.

However, the accuracy of DS is not satisfying. Some variants have been proposed to improve the performance (Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012). They argue that DS introduces a lot of noise into the training data by merging the features of all the sentences containing the same entity pair, because a sentence containing the entity pair of a relation may not talk about the relation. Riedel et al. (2010) and Hoffmann et al. (2011) introduce hidden variables to indicate whether a sentence is noise and try to infer them from the data. Takamatsu et al. (2012) design a generative model to identify noise patterns. However, as shown in the experiments (Section 4), the above variants do not lead to much improvement in accuracy.

In this paper, we point out and analyze some critical factors in DS which have great impact on the accuracy but has not been touched or well handled before. First, each relation has its own schema definition, i.e., the source entity and the destination entity should be of valid types, which is overlooked in DS. Therefore, we propose a component of entity type detection to check it. Second, DS introduces many false negative examples into the training set and we propose a new method to construct negative training examples. Third, we find it is difficult for a single classifier to achieve high accuracy and hence we train multiple classifiers and ensemble them.

We also notice that Nguyen and Moschitti (2011a) and Nguyen and Moschitti (2011b) utilize external information such as more facts from Yago and labeled sentences from ACE to improve the

* The contact author.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 810–815,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

performance. These methods can also be equipped with the approach proposed in this paper.

2 Critical Factors Affecting the Accuracy

DS has four steps: (1) Detect candidate entity pairs in the corpus. (2) Label the candidate pairs using the KB. (3) Extract features for the pair from sentences containing the pair. (4) Train a multi-class classifier. Among these steps, we find the following three critical factors have great impact on the accuracy (see Section 4 for the experimental results).

Valid entity type detection. In DS, a sentence with a candidate entity pair a sentence with two candidate entities is noisy. First, the schema of each relation in the KB requires that the source and destination entities should be of valid types, e.g., the source and destination entity of the relation "DirectorOfFilm" should be of the types "Director" and "Film" respectively. If the two entities in a sentence are not of the valid types, the sentence is noisy. Second, the sentence may not talk about the relation even when the two entities are of the valid types. The previous works (Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012) do not distinguish the two types of noise but directly infer the overall noise from the data. We argue that the first type of noise is very difficult to be inferred just from the noisy relational labels. Instead, we decouple the two types of noise, and utilize external labeled data, i.e., the Wikipedia anchor links, to train an entity type detection module to handle the first type of noise. We notice that when Ling and Weld (2012) studied a fine-grained NER method, they applied the method to relation extraction by adding the recognized entity tags to the features. We worry that the contribution of the entity type features may be drowned when many other features are used. Their method works well on relatively small relations, but not that well on big ones (Section 4.2).

Negative examples construction. DS treats the relation extraction as a multi-class classification task. For a relation, it implies that the facts of all the other relations together with the "Other" class are negative examples. This introduces many false negative examples into the training data. First, many relations are not exclusive with each other, e.g., "PlaceOfBorn" and "PlaceOfDeath", the born place of a person can be also the death place. Second, in DS, the "Other" class is composed of all the candidate entity pairs not existed in the KB, which actually contains many positive facts of non-Other relations because the KB is not complete. Therefore we use a different way to construct negative training examples.

Feature space partition and ensemble. The features used in DS are very sparse and many examples do not contain any features. Thus we employ more features. However we find it is difficult for a single classifier on all the features to achieve high accuracy and hence we divide the features into different categories and train a separate classifier for each category and then ensemble them finally.

3 Accurate Distant Supervision (ADS)

Different from DS, we treat the extraction problem as N binary classification problems, one for each relation. We modify the four steps of DS (Section 2). In step (1), when detecting candidate entity pairs in sentences, we use our entity type detection module (Section 3.1) to filter out the sentences where the entity pair is of invalid entity types. In step (2), we use our new method to construct negative examples (Section 3.2). In step (3), we employ more features and design an ensemble classifier (Section 3.3). In step (4), we train N binary classifiers separately.

3.1 Entity Type Detection

We divide the entity type detection into two steps. The first step, called boundary detection, is to detect phrases as candidate entities. The second step, called named entity disambiguation, maps a detected candidate entity to some entity types, e.g., "FilmDirector". Note that an entity might be mapped to multiple types. For instance, "Ventura Pons" is a "FilmDirector" and a "Person".

Boundary Detection Two ways are used for boundary detection. First, for each relation, from the training set of facts, we get two dictionaries (one for source entities and one for destination entities). The two dictionaries are used to detect the source and destination entities. Second, an existing NER tool (StanfordNER here) is used with the following postprocessing to filter some unwanted entities, because a NER tool sometimes produces too many entities. We first find the *compatible NER tags* for an entity type in the KB. For example,

for the type "FilmDirector", the compatible NER tag of Standford NER is "Person". To do this, for each entity type in the KB, we match all the entities of that type (in the training set) back to the training corpus and get the probability $P_{tag}(t_i)$ of each NER tag (including the "NULL" tag meaning not recognized as a named entity) recognized by the NER tool. Then we retain the top k tags $S_{tags} = \{t_1, \cdots, t_k\}$ with the highest probabilities to account for an accumulated mass z:

$$k = \arg\min_k \left(\left(\sum_{i=1}^{k} P_{tag}(t_i) \right) \geq z \right) \quad (1)$$

In the experiments we set $z = 0.9$. The *compatible ner tags* are $S_{tags} \backslash \{$"NULL"$\}$. If the retained tags contain only "NULL", the candidate entities recognized by NER tool will be discarded.

Named Entity Disambiguation (NED) With a candidate entity obtained by the boundary detection, we need a NED component to assign some entity types to it. To obtain such a NED, we leverage the anchor text in Wikipedia to generate training data and train a NED component. The referred Freebase entity and the types of an anchor link in Wikipedia can be obtained from Freebase.

The following features are used to train the NED component. **Mention Features**: Uni-grams, Bi-grams, POS tags, word shapes in the mention, and the length of the mention. **Context Features**: Uni-grams and Bi-grams in the windows of the mention (window size = 5).

3.2 Negative Examples Construction

Treating the problem as a multi-class classification implies introducing many false negative examples for a relation; therefore, we handle each relation with a separate binary classifier. However, a KB only tells us which entity pairs belong to a relation, i.e., it only provides positive examples for each relation. But we also need negative examples to train a binary classifier. To reduce the number of false negative examples, we propose a new method to construct negative examples by utilizing the 1-to-1/1-to-n/n-to-1/n-to-n property of a relation.

1-to-1/n-to-1/1-to-n Relation A 1-to-1 or n-to-1 relation is a functional relation: for a relation r, for each valid source entity e_1, there is only one unique destination entity e_2 such that $(e_1, e_2) \in r$. However, in a real KB like Freebase, very few relations meet the exact criterion. Thus we use the following approximate criterion instead: relation r is *approximately* a 1-to-1/n-to-1 relation if the Inequalities (2,3) hold, where M is the number of unique source entities in relation r, and $\delta(\cdot)$ is an indicator function which returns 1 if the condition is met and returns 0 otherwise. Inequality (2) says the proportion of source entities which have exactly one counterpart destination entity should be greater than a given threshold. Inequality (3) says the average number of destination entities of a source entity should be less than the threshold. To check whether r is a 1-to-n relation, we simply swap the source and destination entities of the relation and check whether the reversed relation is a n-to-1 relation by the above two inequalities. In experiments we set $\theta = 0.7$ and $\gamma = 1.1$.

$$\frac{1}{M} \sum_{i=1}^{M} \delta\left(\left| \{e' | (e_i, e') \in r\} \right| = 1 \right) \geq \theta \quad (2)$$

$$\frac{1}{M} \sum_{i=1}^{M} \left| \{e' | (e_i, e') \in r\} \right| \leq \gamma \quad (3)$$

n-to-n Relation Relations other than 1-to-1/n-to-1/1-to-n are n-to-n relations. We approximately categorize a n-to-n relation to n-to-1 or 1-to-n by checking which one it is closer to. This is done by computing the following two values α_{src} and α_{dst}. r is treated as a 1-to-n relation if $\alpha_{src} > \alpha_{dst}$ and as a 1-to-n relation otherwise.

$$\begin{aligned} \alpha_{src} &= \frac{1}{M_{src}} \sum_{i=1}^{M_{src}} \left| \{e' | (e_i, e') \in r\} \right| \\ \alpha_{dst} &= \frac{1}{M_{dst}} \sum_{i=1}^{M_{dst}} \left| \{e' | (e', e_i) \in r\} \right| \end{aligned} \quad (4)$$

Negative examples For a candidate entity pair (e_1, e_2) not in the relation r of the KB, we first determine whether it is 1-to-n or n-to-1 using the above method. If r is 1-to-1/n-to-1 and e_1 exists in some fact of r as the source entity, then (e_1, e_2) is a negative example as it violates the 1-to-1/n-to-1 constraint. If r is 1-to-n, the judgement is similar and just simply swap the source and destination entities of the relation.

3.3 Feature Space Partition and Ensemble

The features of DS (Mintz et al., 2009) are very sparse in the corpus. We add some features in (Yao et al., 2011): **Trigger Words** (the words on the dependency path except stop words) and **Entity String** (source entity and destination entity).

Relation	Taka	Ensemble
works_written	0.76	0.98
river/basin_countries	0.48	1
/film/director/film	0.82	1
Average	0.79	0.89

Table 1: Manual evaluation of top-ranked 50 relation instances for the most frequent 15 relations.

We find that without considering the reversed order of entity pairs in a sentence, the precision can be higher, but the recall decreases. For example, for the entity pair ⟨Ventura Pons, Actrius⟩, we only consider sentences with the right order (e.g. *Ventura Pons is directed by Actrius.*). For each relation, we train four classifiers: C_1 (without considering reversed order), C_2 (considering reversed order), C_{1more} (without considering reversed order and employ more feature) and C_{2more} (considering reversed order and employ more feature). We then ensemble the four classifiers by averaging the probabilities of predictions:

$$P(y|x) = \frac{P_1 + P_2 + P_{1more} + P_{2more}}{4} \quad (5)$$

4 Experiments

4.1 Dataset and Configurations

We aimed to extract facts of the 92 most frequent relations in Freebase 2009. The facts of each relation were equally split to two parts for training and testing. Wikipedia 2009 was used as the target corpus, where 800,000 articles were used for training and 400,000 for testing. During the NED phrase, there are 94 unique entity types (they are also relations in Freebase) for the source and destination entities. Note that some entity types contain too few entities and they are discarded. We used 500,000 Wikipedia articles (2,000,000 sentences) for generating training data for the NED component. We used Open NLP POS tagger, Standford NER (Finkel et al., 2005) and MaltParser (Nivre et al., 2006) to label/tag sentences. We employed liblinear (Fan et al., 2008) as classifiers for NED and relation extraction and the solver is L2LR.

4.2 Performance of Relation Extraction

Held-out Evaluation. We evaluate the performance on the half hold-on facts for testing. We compared performance of the $n = 50,000$ best extracted relation instances of each method and the Precision-Recall (PR) curves are in Figure 1 and Figure 2. For a candidate fact without any entity existing in Freebase, we are not able to judge whether it is correct. Thus we only evaluate the candidate facts that at least one entity occurs as the source or destination entity in the test fact set.

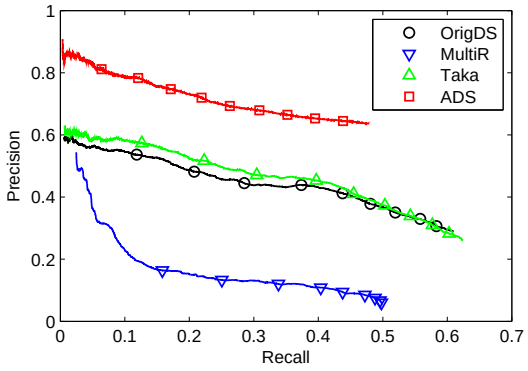

Figure 1: Performance of different methods.

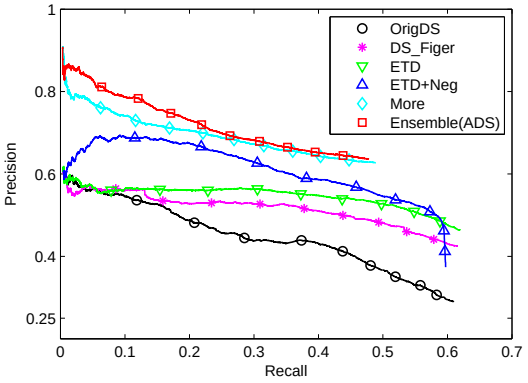

Figure 2: Contributions of different components.

In Figure 1, we compared our method with two previous methods: MultiR (Hoffmann et al., 2011) and Takamatsu et al. (2012) (Taka). For MultiR, we used the author's implementation[1]. We re-implemented Takamatsu's algorithm. As Takamatsu's dataset (903,000 Wikipedia articles for training and 400,000 for testing) is very similar to ours, we used their best reported parameters. Our method leads to much better performance.

Manual Evaluation. Following (Takamatsu et al., 2012), we selected the top 50 ranked (according to their classification probabilities) relation facts of the 15 largest relations. We compared our results with those of Takamatsu et al. (2012) and we achieved greater average precision (Table 1).

[1]available at http://www.cs.washington.edu/ai/raphaelh/mr We set $T = 120$, which leads to the best performance.

P_{micro}	R_{micro}	P_{macro}	R_{macro}
0.950	0.845	0.947	0.626

Table 2: Performance of the NED component

4.3 Contribution of Each Component

In Figure 2, with the entity type detection (ETD), the performance is better than the original DS method (OrigDS). As for the performance of NED in the Entity Type Detection, the Micro/Macro Precision-Recall of our NED component are in Table 2. ETD is also better than adding the entity types of the pair to the feature vector (DS_Figer)[2] as in (Ling and Weld, 2012). If we also employ the negative example construction strategy in Section 3.2 (ETD+Neg), the precision of the top ranked instances is improved. By adding more features (More) and employing the ensemble learning (Ensemble(ADS)) to ETD+Neg, the performance is further improved.

5 Conclusion

This paper dealt with the problem of improving the accuracy of DS. We find some factors are crucially important, including valid entity type detection, negative training examples construction and ensembles. We have proposed an approach to handle these issues. Experiments show that the approach is very effective.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 2, pages 3–3.

[2]We use Figer (Ling and Weld, 2012) to detect entity types

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*. 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 Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June. Association for Computational Linguistics.

X. Ling and D.S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel 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 International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August. Association for Computational Linguistics.

Truc-Vien T. Nguyen and Alessandro Moschitti. 2011a. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 277–282, Stroudsburg, PA, USA. Association for Computational Linguistics.

Truc-Vien T Nguyen and Alessandro Moschitti. 2011b. Joint distant and direct supervision for relation extraction. In *Proceeding of the International Joint Conference on Natural Language Processing*, pages 732–740.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *In Proc. of LREC-2006*, pages 2216–2219.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML-2010)*, pages 148–163.

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 (Volume 1: Long Papers), pages 721–729, Jeju Island, Korea, July. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK., July. 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 of the Association for Computational Linguistics (ACL'05)*, pages 419–426, Ann Arbor, Michigan, June. Association for Computational Linguistics.