Distant Supervision for Relation Extraction

Zhen Wang (v-zw)

October 29, 2013

1 Introduction

Knowedge base (KB) is a systematical way to organize worldly facts together which is proved to be very helpful for many natural language processing (NLP) applications such as question answering (QA). The state-of-the-art KB such as Freebase maintains millions of *entities* (individuals or objects) and thousands of kinds of *relations* between entities. Currently, the relations in which we are interested are mainly binary relations such as 'CEO-of', 'Born-in', 'Contains', etc. Binary relations are the building blocks of more complex relationships. Regarding each entity as a node and each binary relation as one kind of directed edge, the node 'Steve Jobs' and the node 'Apple' are connected by an edge of kind 'CEO-of' in Freebase. We say that the entity pair '(Steve Jobs, Apple)' is an *relation instance* of the relation 'CEO-of'.

To construct an open domain KB which provides many useful prior for downstream applications, manually annotating a lot of relation instances is not practical. As the amount of electronic text corpora becomes larger and larger, discovering structured information from unstructured natural language attracts more and more attention from both academic and industry communities. Many approaches have been proposed to explore the possibility of automatically augmenting or even constructing a KB from a given text corpus. Since there have been several KB which are comparatively large, some efforts have been devoted to learning new facts from KB itself. In summary, the task of *Relation extraction* have the following different scenarios:

- relation set
  - only pre-defined relations are considered
  - open information extraction (IE): to discover unseen relations which are explicitly expressed by a surface string or characterized by a collection of related instances
- entity set
  - only existed entities are considered
  - unseen entities are detected from some source and taken into further consideration
- source of novel facts
  - discovering novel relation instances from a text corpus
  - predicting missing edges in existed KB to augment it with new facts
- which element to be predicted, i.e., considering an relation instance as a triple (e, p, v) where p is the predicate (relation), the form of problem is
  - relation prediction: given (e, v) to guess p
  - slot filling (i.e., one sub-task of knowledge base population (KBP)): given $e, p)$ to guess v

In this survey, we mainly consider the scenario of harvesting many unseen instances of specific relations from a text corpus, say that in our setting, unseen entities are detected and treated as arguments of relations but we only try to extract facts for pre-defined relations.

As a sub-task of information extraction, most previous works assume that there is an oracle which is capable to detect named entities from input text snippets and classify each entity to one of pre-defined classes (i.e., entity types). Then they focus on judging whether an entity pair posseses some relation. Like most classification problems, existing approaches can be divided into the following three genres:

- supervised learning: A training set is manually annotated where an observation is an entity pair as well as its feature representation and label is the relation connecting the two entities.
  - advantage: clean training data means ensurance of precision
  - disadvantage: building such a training set is costly and not practical for scaling up. Thus it is limited to small dataset or specific domain
- bootstrapping: A small number of seed examples are carefully generated and used to train a model. Then new relation instances are discovered by current model and examples with high confidence are added to seed set for re-training. This procedure is repeated iteratively.
  - advantage: significantly reducing the number of training examples required in pure supervision case
  - disadvantage: the iterative process may lead to semantic-drift
- unsupervised learning: aggregates similar relation instances together by discovering the intrinsic strucrue within their feature space
  - advantage: can be applied to very large text corpora
  - disadvantage: it is difficult to map each cluster of relation instances to a canonical realtion type defined in existed KB and thus novel relation instances can not be immediately inserted into KB
- Distant supervision (DS) [14]: Provdies a paradigm to heuristically label a large corpus without human labor for training a relation extractor.

In this survey, we focus on distant supervision for relation extraction. We recommand Bach et al [3] to readers who are interested in a comprehensive review of relation extraction.

2 Distant Supervision for Relation Extraction

DS [14] heuristically labels data based on a simple assumption that given

- Σ, a large corpus e.g., Wikipedia articles, NYT, etc.
- E, a set of entities mentioned in Σ e.g., 'Barack Obama', 'Gone with the wind', etc.
- R, a set of relation types e.g., 'CEO-of', 'Born-in', etc.
- Δ, a set of ground relation instances of relations in R e.g., 'CEO-of(Steve Ballman, Microsoft)', 'Born-in(Barack Obama, Ohio)', etc.
- T, a set of entity types e.g., 'PER', 'ORG, 'LOC', etc, as well as type signature $r(E_1, E_2)$ for relations e.g., 'Born-in(PER, LOC)'.

for any sentence $s \in \Sigma$ that contains a pair of entity (e_1, e_2), if $\exists r \in R$ s. t. $r(e_1, e_2) \in \Delta$, regard s as an expression of relation r.

As a running example, for the sentence 'Obama was born in Ohio', they use Stanford's named entity tagger to detect all named entities $\{$'Obama', 'Ohio'$\}$ in the sentence. For each entity pair i.e., '(Obama, Ohio)' in this case, suppose 'CEO-of(Obama, Ohio)' is a ground relation instance in Freebase, they extract both lexical and syntactic features from this sentence w.r.t. '(Obama, Ohio)'. A typical lexical feature is the sequence of both tokens and part-of-speech (POS) tags between the two entities, i.e., 'was/V born/V in/PP' in this example. A typical Syntactic feature is the seqence of dependency path between the two entities. Finally, they combine features extracted from *all* mentions of '(Obama, Ohio)' to form a positive example—(x_i =features, y_i ='CEO-of'). if '(Obama, Ohio)' is not a instance of any relation in Freebase, a negative example is constructed in a similar way.

The larger a text corpus is, the larger the training set generated by DS will be. Although this promising property has made DS for relation extraction one of the hottest research topics, it still posseses some challenging problems. In this survey, we will analysis these problems and discuss related works that made efforts to solve these problems.

3 Challenging Problems

3.1 Named Entity Recognition (NER)

Firstly, the assumption of the existence of a perfect NER tagger does not make sense. Most of previous works made use of the Stanford NER tagger with a 4-classes label set $\{$'PER', 'ORG', 'LOC', 'MISC', 'NONE'$\}$. I thought up 30 entities (each category has 10 entities) include 'Hong Kong', 'Los Angeles', 'Opec', 'NCAA', 'Nixon' and 'Bill Gates', etc. Then I checked their NER tags within each mention of them. According to my observation, the Stanford NER tagger can classify entities without ambiguity into their appropriate classes with high accuracy (about 92%). However, there are many ambiguous names which refer to different entities under different contexts. For example, 'Barcelona' may be a city or a football club where the corresponding NER tag should be 'LOC' or 'ORG'. Within 72 mentions of 'Barcelona' where the NER tags are labeled as 'ORG', only 35 of them are indeed talking about that famous football club. On the other hand, there are 65 mentions in which 'Barcelona' refers to the city in Spain but only 28 of them are tagged as 'LOC'. Similar cases are ubiquitous.

Duing to this reason, original DS for relation extraction align KB to text corpus without type checking (i.e., they do not check whether the NER tags of (e_1, e_2) is consistent with the type signature $r(E_1, E_2)$). However, all the lexical and syntactic features include the NER tags as part of them and thus bring in noisy patterns. For instance, if 'Barcelona' is tagged with 'ORG' in the sentence 'We went to Barcelona, Spain' and the ground relation instance 'Contained(Barcelona, Spain)' is aligned to this sentence, Features collected for 'Contained' will include the incorrect entity type (i.e., 'ORG') which is definitely the noise. Although type signature is a strong and reliable constraint for a relation, those noisy patterns dilute the discriminative capability of it.

I also checked the mentions of entity pairs that are predicted as 'Contains(LOC, LOC)' by classifier but are actually instances of other relation in adopted KB. Within these mentions, most of names refering to a movie or a book are incorrectly tagged as 'LOC'. I think that Stanford NER tagger is a little conservative to announce 'MISC' which possesses the largest number of ambiguous names and is the bottleneck of NER task in my opinion. Someone may find it helpful to improve NER with relation information. For instance, suppose the relation instance 'Act-in(Bogart, Casablanca)'$\in \Delta$ and it can be aligned to a sentence, then we know that 'Casablanca' in this sentence refers to a movie instead of a city. This observation tells us that relation information is beneficial for NER. Another issue related to entity types are their granulity. There are some samll relations (e.g., 'Director-of', 'Writer-of') which may benefit from a fine-grained entity types (e.g., 'Director', 'Writer') because only a small fraction of entities belong to 'PERSON' are actually 'Director'/'Writer'. Training a tagger which predicts fine-grained entity types is helpful for improving the performance of relation extraction but it is difficult to ensure the tagger's accuracy when the number of classes (entity types) increases.

3.2 Quality of Training Data

As a weak supervision framework, the quality of training data which is heuristically labeled is not as high as the quality of manually annotated training data. As you can see, original assumption results both false positive examples and false negative examples.

When two different relations share a common entity pair, each sentence contains this entity pair will generate positive examples for both the two relations. However, each sentence (mention) expresses one of the two relations. For example, 'Michael Jackson gave a show in Gary.' expresses 'Work-at(Michael Jackson, Gary)' but it will be regarded as an example of relation 'Born-in' because 'Born-in(Michael Jackson, Gary)'$\in \Delta$. This problem is serious if the overlapping between different relations is large to some extent. When I applied Jaccard similarity to measure the overlapping between different relations of Freebase, the largest Jaccard similarity value 0.22 is achieved by 'people.place-lived.location' and 'person.place-of-birth'. In such a case, our classifier must not be able to distinguish the two relations accurately. Even when an entity pair is uniquely possessed by only one relation, a mention of it may express a certain relation which is not taken into our consideration. As a multi-class classification problem, these noisy patterns should be absorbed into 'OTHER' class (i.e., as negative example). The two reasons that causes false positive examples result from the KB itself. Hence, it is extremely challenging to solve them.

As for negative examples, this phenomenon is caused by incompleteness of adopted KB. Although the size of the-state-of-the-art KB like Freebase has became larger and larger, it still only covers part of adopted text corpus. In the latest Freebase, 'Location.contains' has about $75,687,676$ relation instances. When we align them to sentences of Wiki articles, only about $50,999$ relation instances are matched.

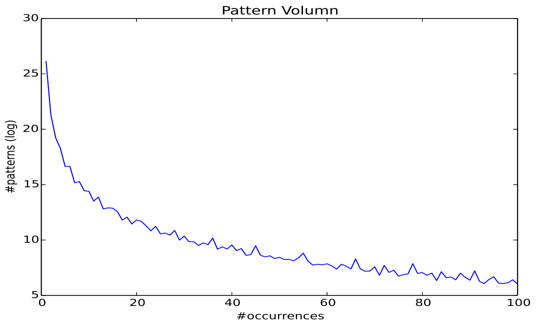

Figure 1: The Distribution of Pattern Volumn

However, in our held-out evaluation, when I checked the top 100 entity pairs that are incorrectly-labeled as 'location.contains' (entity pairs which are predicted as 'Location.contains' but not appear in held-out instances of this relation), I found that most of them are actually facts of our world. This observation reflects that even though Freebase increases its size at a more rapid pace than Wikipedia, the size of their intersection is limited and there are still many facts in Wiki articles that are out of Freebase. In a word, original DS's method for constructing negative examples suffers from the possibility of generating false negative examples. Besides, the training data is extremely unbalanced. Even only one false negative example may be fatal for some small (unpopular, specific) relations because of their limited numbers of positive examples.

3.3 Sparseness of Feature Space

A critical problem is the sparse feature space. Most of related works made use of several lexical and syntactic features which are the conjunction of several attributes of the context w.r.t. an entity pair. These features are extremely discriminative and results satisfactory precision. In original DS's experiment, they report $400,000$ as the number of distinct features in their logistic regression classifier. In our experiment, the number of features without filtering low document frequency ones is about one million and the distribution of features' volumn reveals the long tail property (See Figure 1). In essense, each relation is characterized by a set of features (patterns or say expressions) collected through DS from given text corpus. To express the semantics of a relation between two entities, human beings can think up infinite expressions but the number of mentions in our training set is finite and thus it is impossible for collected expressions to cover all the possible expressions. With those strict, binary features which have no good generalization properties, our model is determined to miss a lot of novel relation instances. As an example, in my held-out evaluation, the number of held-out instances of relation 'Location.contains' that appear in testing corpus is about $31,643$. After filtering out instances whose features have no intersection with features of training data, the number of remaining instances decreases to about $17,192$. This observation tells us one of the reasons of low recall.

4 Related Work

Mike et al [14] proposed the paradigm of DS for relation extraction which opened one door for this research topic even though the method itself is quite simple and has many problems to be addressed.

4.1 Fine-grained Entity Types

According to previous analysis, a fine-grained NER tagger is helpful for disambiguating several possible relations of a mention. As Xiao et al [12] pointed, the main challenges are

- selection of a tag set,
- automatically generating a training set,
- a fast, accuracy classifier.

They proposed an approach to tag each detected named entity with a subset of the universal entity type set (multi label). In their approach, the number of entity types is 112. The 112 entity types are Freebase types that contain at least 5 instances (entities). In contrast to entity type set which is designed by experts, directly making use of Freebase's entity types not only gives a collection of fine-grained entity types without human labor but also leverages Freebase's hierarchical relationships between these entity types. To automatically generate a training set, they adopted a weak supervision approach. Specifically, the *anchors* of Wiki articles provide not only the segmentation of named entities but also their corresponding types defined in Freebase. They directly made use of the anchors of Wiki articles to train a CRF model for segmentation and a multi-class multi-label classifier for assiging NER tags. Besides of traditional features for NER, they also adopted the ReVerb patterns [7] which are verb phrases indicating verbal relations. From this respect, they imported relation information for NER. Zhang et al [21] is the first one to apply the fine-grained entity types for relation extraction task. However, their features are traditional mention features and context features but the relation information has not been made full use.

4.2 Filtering Labeling Errors

Many works mainly focus on improving the quality of training data heuristically labeled by DS. One genre is giving more detailed or coarse representation of patterns to re-estimate the association between patterns and relations expecting that some false positive examples will be removed from a relation's pattern collection and some false negative will be explained by some considered relation due to the co-occurrences (or similarity) between patterns [17, 20, 1, 18]. Another genre integrates this intuition into original discriminative model by multi-instance learning [15, 9, 16, 13].

4.2.1 Representation of Pattern

In essence, DS provides a way to collect difference expressions (patterns) for each specific relation. However, as strict binary features, it is impossible to compare distinguishing features (patterns), or say calculate similarity between them. If patterns are represented in a form which enables comparison between them (e.g., divide original conjunctions into individual attributes of sentences), even though there is no supervision, a generative model can capture the intrinsic structure of feature space and re-estimate the association between features (patterns) and relations. In this way, some labeling errors may be revised so that the quality of training data is improved.

Those generative models seem complex but the underling intuition is very straight. For instance, in Takamatsu et al [17]'s model, both relations and patterns are characterized by their associated entity pairs. Specifically, let's denote the entity pairs of relation r as E_r, the entity pairs of a pattern s as E_s. Intuitively speaking, if we rank a relation r's associated patterns by $\frac{|E_s \cap E_r|}{|E_s|}$ in descending order, the top-ranked ones are more likely to express r. The more similar a pattern's entity pairs are w.r.t top-ranked patterns' entity pairs, the more likely for it to express r and vice versa. Representing a pattern in such a manner allows us to compare different patterns. Another benefit is that although entity pairs are ambiguous in many cases, a pattern expresses only one meaning. Yao et al [20] treats a document as a distribution over relations. When a relation r is generated, it is responsible for explaining the features, source entity's type, destination entity's type. Then the entity type specific distribution is used to sample observed entities. This generative story makes a lot of sense but in my opinion, the most important contribution is breaking down original patterns into fine-grained features which are explained by a relation respectively. Patterns which are regared as independent objects in original DS now can be compared, or say that patterns are represented as fine-grained features and thus co-occurrences of these features contribute to similarity between patterns.

To reduce false nagative examples, Wei et al [18] proposed to leverage passage retrieval and psuedo relevance feedback with coarse features (patterns). Firstly, sentences w.r.t. a relation r are divided into three categories:

- $POS(r) = \{s \in \Sigma | s \text{ contains some } (e_1, e_2) \text{ where } r(e_1, e_2) \in \Delta\}$
- $RAW(r) = \{s \in \Sigma | s \text{ contains some } (e_1, e_2) \text{ satisfing } r(E_1, E_2)\}$
- $NEG(r) = \{s \in \Sigma | s \text{ contains some } (e_1, e_2) \text{ satisfying } r(E_1, E_2) \text{ and } \exists e' \text{ s.t. } r(e_1, e') \in \Delta \text{ and } r(e_1, e_2) \notin \Delta\}$

Then a passage retrieval component (given a relation r as query, retrieve sentences that are most likely to express r) is trained with positive examples $POS(r)$ and negative examples $NEG(r)$. They use this component to select "relevant" passages (i.e., sentences) from $RAW(r)$ and then select entity pairs from these top-ranked sentences. Selected entity pairs are added into KB to alleviate its incompleteness. When a relation r is a 1-to-n relation and our KB misses $r(e_1, e_2)$, the principle for constructing $NEG(r)$ is still likely to include false negative examples.

4.2.2 Multi-instance Learning

As for multi-instance learning genre, they are based on the relaxed assumption that if $\exists r(e_1, e_2) \in \Delta$, *at least one* of (e_1, e_2)'s mentions express r. In most cases, this kind of models for relation extraction have a hidden layer of random variables to model the mention level labels and a layer of random variables to model entity level labels which are influenced by mention level labels in a aggregation manner. The entity level is used to inject supervision into the model. To reflect the 'at-least-one' assumption, Riedel et al [15] proposed to model the event that whether the i-th mention of an entity pair expresses a relation r by a binary hidden variable Z_i. Hoffmann et al [9] proposed the *MultiR* which is used as baseline in many related works. In their model, each mention is associated with a random variable taking value from $R \cup \{\text{OTHER}\}$. Then entity level labels are modeled in a binary vector random variable $\mathbf{Y}$ where $Y^r = 1$ means this entity pair has relation r and vice versa. The relationship between hidden layer variables (mention level) and entity level variable is simply a deterministic-or instead of any complex conditional probability distribution. The deterministic-or relation is beneficial for simplifing inference procedure but is not able to describe the scenario where appearrance of r at mention level frequently co-occurs with appearrance of r' at entity level. Surdeanu et al [16] proposed a three layer model which seems similar with MultiR but given an observation (a feature representation of a mention), it predict which relation this mention expresses by a logistic regression classifier and then use $|R|$ logistic regression classifier to predict entity level labels according to all the mention level labels respectively. In such case, a mention level prediction in which only 'Born-in' is activated may also have opportunity to trigger 'Live-in' label at entity level if for most entity pairs, there exists a strong correlation between them. Min et al [13] advance further to seperate entity level labels into two layers, one as hidden layer modeling the predictions, one as observed supervision. By allowing an inconsistent states between the hidden layer of entity level and supervision, an entity pairs may be associated with some relation which is supported by mention level labels rather than existed ground truth provided by KB which seems helpful for alleviating the influences made by incompleteness of KB. Takamatsu et al. [**?**] pointed out that 91.7% entity pairs appear only once in Wiki articles. In such case, the 'at-least-one' assumption is equivalent to original DS assumption for these multi-instance learning approaches.

4.3 Embedding

Up to now, all these approaches discussed made use of strict binary features which lack of local smooth property and thus can't generalize well. Each relation is characterized by a collection of such features. It is definitely a bad representation of a specific relation. To tackle the sparseness of feature space, word embedding related techniques are imported to relation extraction during the last two years [6, 19, 5].

Most of traditional dimension reduction approaches fit a common pipeline where a similarity matrix is generated firstly and then factorized to select now basis. For a graph, the similarity matrix is usually the connectivity between nodes. However, in relation extraction, both people are interested in both nodes (entities) and edges (relations). Thus, how to embed a relation in a reasonable way is the most critical point. A straightforward idea is to represent each relation by a linear mapping (matrix) from a source entity to a destination entity. Under this hyperthesis, the constraints used to estimate parameters (entries of a matrix) is usually minimizing $\|\mathbf{v} - R\mathbf{e}\|$. Naturally, quadratic form is also explored in which a true relation instance (e_1, e_2) is assigned a higher score $g_R(\mathbf{e}_1, \mathbf{e}_2) = \mathbf{e}_1^{\mathrm{T}} R \mathbf{e}_2$ than a false instance. Furthermore, Chen et al [6] propose to represent an entity by the average of its words' vector representation, a relation by the parameters of a neural tensor network. Given an entity pair (e_1, e_2), a score of how plausible they are in a certain relationship r is computed by $g(e_1, r, e_2) = U^{\mathrm{T}} tanh\{e_1^{\mathrm{T}} W_r^{1:k} e_2 + V_r \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_r\}$ where $e_1^{\mathrm{T}} W_r^{1:k} e_2$ results a k-dimensional vector h where $h_i = e_1^{\mathrm{T}} W_r^i e_2$. To estimate the parameters describing a certain relation r, they acquire supervision directly from adopted KB. The intuition is quite simple, any ground triple in KB should achieve a larger score than any invalid triple which is constructed by fixing $e1, r$ and randomly selecting a corrupted entity e_c.

Weston et al [19] also leverage word embedding related technique especially a reasonable geometrical explaination of how a relation connects two entities which seems more reasonable than previous embedding models. In their model, constraints for parameter estimation are drawn from both DS and KB. Firstly, a mention-relation pair is scored by $S_{m2r}(m, r) = \mathbf{f}(m)^{\mathrm{T}}\mathbf{r}$ where $\mathbf{f}(m) = W^{\mathrm{T}}\mathbf{\Phi}(m)$ and $\mathbf{r}$ is the embedding of relation r. The matrix W consists of all the distributed representations of words and $\mathbf{\Phi}(m)$ is the binary representation of m (a mention is a size-k window here). To estimate the parameters, DS heuristically labels mention-relation pairs. Then the constraints are $\forall i, \forall r' \neq r_i, \mathbf{f}(m_i)^{\mathrm{T}}\mathbf{r}_i > 1 + \mathbf{f}(m_i)^{\mathrm{T}}\mathbf{r}'$ The most important point proposed in their work is to regard a realtion r as a *translation* from source entity's vector $\mathbf{h}$ to destination entity's vector $\mathbf{t}$. This intuitive geometric explaination naturally inspired them to minimize $S_{kb}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$. Specifically, they adopted a ranking loss, say that appling $S_{kb}()$ to any ground truth triple should result a score which is larger than one plus a score generated by appling $S_{kb}()$ to a corrupted triple. However, the translation based intuition doesn't make any sense for 1-to-n (or n-to-1) relations and symmetrical relations like 'Brother-of'. Finally, we have to point that a translation dominated by a k-dimensional vector can be expressed by a linear mapping characterized by a $k+1$-dimensional matrix.

4.4 Miscellaneous

4.4.1 Knowledge Base Population (KBP)

I also surveyed works on KBP [11, 8, 2]. The KBP contest is so relevant with relation extraction. It mainly consists of two sub-tasks:

- Entity Linking: given a name, decide whether this name corresponds to an entity in a database and, if so, which one.
- Slot Filling: given an entity and a attribute of it, determine from a large corpus the values of the specific attribute (some attributes are multi-valued).

Heng et al [11] gave an overview of TAC 2010 KBP track. For entity linking, they pointed that

- unsupervised learning can achieve comparable performance as supervised learning.
- semantic features such as synonys and variants are helpful.
- Wikipedia's structure such as anchors, redirects and disambiguatioin should be made full use.

For slot filling, they gave the following points:

- coreference and cross-slot inference are necessary because 39.6% answers require system to go beyond single sentence extraction and attribute has many different expressions (e.g., 'son' is 'child').
- external knowledge, although extremely huge covers limited slot fillers (consistent with my observation).

For my best knowledge, Chen et al [2] is the first one to emphasize that the slot filling results as feedback are helpful for improving the entity linking task. Besides, they gave an interesting approach for integrating these two tasks together. Each entity node as well as the given query is characterized by its profile which is defined as a list of attributes as well as their value (s). The value (s) of attributes of an entity node are harvested by slot filling.

4.4.2 Wikipedia's Structures

Most approaches on relation extraction process tens of relations. Hoffmann et al [10] proposed a method to collect relation instances for about 5000 relations. Although their setting is not identical as traditional relation extraction, preparing rich lexicon to calculate lexicon features and leveraging info-box within Wiki articles to provide relation (attribute or predicate) schema of a document (article) are inspiring ideas for future works.

4.4.3 Open Information Extraction

Open information extraction (Open IE) does not require specific relations as input and automatically discovers novel relations as well as words which are indicative of corresponding relations. Banko et al [4] gave an important observation that only several lxico-syntactic pattern can cover most (about 95%) sentences which express the meaning of a certain binary relation). Inspired by this observation, Fader et al [7] build the-state-of-the-art open IE system—ReVerb. The differences deserve our attention are:

- For relations which are usually expressed as a verbal function, those lexico-syntactic pattern should be integarted into our model. We believed that these patterns have the potential to standardize or validate extracted features.
- In traditional relation extraction task, entities are firstly detected and features are extracted w.r.t. an detected entity pair. In contrast, ReVerb firstly locates the verb phrases of relations and then considers nearest entities as arguments.

5 Proposal

References

[1] E. Alfonseca, K. Filippova, J.-Y. Delort, and G. Garrido. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 54–59. Association for Computational Linguistics, 2012.

[2] J. Artiles, Q. Li, T. Cassidy, S. Tamang, and H. Ji. Cuny blender tackbp2011 temporal slot filling system description. In *Proceedings of Text Analysis Conference (TAC)*, 2011.

[3] N. Bach and S. Badaskar. A review of relation extraction. *Literature review for Language and Statistics II*, 2007.

[4] M. Banko, O. Etzioni, and T. Center. The tradeoffs between open and traditional relation extraction. In *ACL*, volume 8, pages 28–36, 2008.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Irreflexive and hierarchical relations as translations. *arXiv preprint arXiv:1304.7158*, 2013.

[6] D. Chen, R. Socher, C. D. Manning, and A. Y. Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*, 2013.

[7] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[8] B. M. X. L. R. Grishman and A. Sun. New york university 2012 system for kbp slot filling.

[9] R. Hoffmann, C. Zhang, X. Ling, L. S. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550, 2011.

[10] R. Hoffmann, C. Zhang, and D. S. Weld. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics, 2010.

[11] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, 2010.

[12] X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*, 2012.

[13] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of NAACL-HLT*, pages 777–782, 2013.

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

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

[16] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. 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*, pages 455–465. Association for Computational Linguistics, 2012.

[17] S. Takamatsu, I. Sato, and H. Nakagawa. 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*, pages 721–729. Association for Computational Linguistics, 2012.

[18] X. Wei, R. H. Le Zhao, and R. Grishman. Filling knowledge base gaps for distant supervision of relation extraction.

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

[20] L. Yao, A. Haghighi, S. Riedel, and A. McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

[21] X. Z. J. Zhang, J. Z. J. Y. Z. Chen, and Z. Sui. Towards accurate distant supervision for relational facts extraction.