Neural Computing and Applications
https://doi.org/10.1007/s00521-018-3453-x

S.I. : EMERGENCE IN HUMAN-LIKE INTELLIGENCE TOWARDS CYBER-PHYSICAL SYSTEMS

Multiple order semantic relation extraction

Shengli Song[1,2] · Yulong Sun[1] · Qiang Di[2]

Received: 26 December 2017 / Accepted: 23 March 2018

Abstract
In order to get more comprehensive and accurate knowledge from the Semantic Web, it is essential to design an effective method to extract semantic data from the web and documents. However, as a crucial topic of semantic information extraction, relation extraction is a major challenge in knowledge base construction. Existing methods perform poorly on texts in the open domain due to their complex structure; this paper proposes a method named multiple order semantic relation extraction (MOSRE), which applies for multiple orders, a conceptual expression used in formal logistics, to build semantic patterns for extracting information from hybrid unstructured texts in the open domain with deep semantic analyses. Specifically, the proposed method automatically constructs a multiple order semantic tree from complex natural sentences and converts semantic information into a binary structure. Instead of constructing a large amount of pattern sets for comparing binary semantics between entities, MOSRE splits and reconstructs sentences into a strict hierarchical binary structure with combination rules in order to extract as much semantic information as possible. The extracted triples are then processed into several entity relations in the format ⟨*subject, relational label, object*⟩ after named entity recognition and refinement. MOSRE is validated by test results on two different datasets, achieving $F1$ values of 83.8% on SENT500 and 35.5% on KBP.

Keywords Semantic Web · Relation extraction · Multiple order semantic parsing · Hierarchical binary structure

1 Introduction

Knowledge base construction (KBC), the process of populating a knowledge base with semantic information from structured and unstructured text, is a significant and difficult task. In recent years, several large-scale knowledge bases have been constructed, such as YAGO [29], NELL [30], DBpedia [31], IBM's Watson [32] and Microsoft's EntityCube [27]. However, most of these knowledge bases are generated from semi-structured data and structured data, and the rapid growth of unstructured information on the web is driving the development of methods that enable information extraction from unstructured data to build a knowledge base.

Relation extraction (RE) is an important but unsolved problem in knowledge base construction. Since RE has been proposed, the field has been focused on the automatic extraction of structured relationships from unstructured sources such as documents, blogs and webs, which can potentially benefit a wide range of natural languages processing tasks, such as question answering, ontology learning and summarization. At the time of writing, many intelligent systems are developed based on the knowledge base, which urges the research of relation extraction in the open domain.

In order to address the challenge of relation extraction of complex sentences, various approaches have been proposed. Open information extraction (Open IE) proposed by [5] has been used for relation extraction. Many related Open IE-based systems can extract informal relational information by manually or generated patterns set, such as REVERB [10], ClausIE [7], ReNoun [23]. Additionally,

✉ Shengli Song
shlsong@xidian.edu.cn

1 Software Engineering Institute, Xidian University, Xi'an 710071, China

2 School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Published online: 26 April 2018

there are many approaches using machine learning techniques that have been proposed to solve this problem. After learning many patterns and features in corresponding instances of the relation label, these methods try to extract relation triples as much as possible. From the research and standard assessment methods in recent years, there are also some major problems to be resolved.

Multiple surface patterns The complex sentences contain various surface patterns between entity pairs. Semantic relation information is hidden in some nested or related structures. This results in greater complexity of lexical and syntactic features.

A large amount of noise There are often multiple subjects, objects and paths in a complex sentence. Not all the paths between entity pairs imply semantic relations. These unrelated entity pairs may be interpreted as noise during the process of relation extraction, which means the relation extraction is not only a classification task.

This paper proposes a novel approach to multiple order semantic relation extraction (MOSRE) to extract semantic information, especially multiple order semantics, from complex and compound sentences. MOSRE splits sentences and reorganizes them into a multilayer binary tree for extracting semantic information in triple format. After that, these semantic triples are assembled into relational triples between entities by name entity recognition and refinement.

The main contributions in this paper include: (1) We define a novel concept of multiple order semantic to describe the structure of multiple semantics in the compound-complex sentence. (2) We propose a method of multiple order semantic parsing, which extracts patterns of the multiple order semantic from complex structure. (3) We apply our proposed method to the RE task, achieving multiple-entity relation extractions from compound-complex sentences. (4) Experiments on two different datasets show that: (1) over 75% of the relational triples in SENT500 have been extracted with 90% precision and (2) over 25 kinds of the relational triples in KBP have been extracted with 55% precision.

2 Related works

Since the task of RE has been proposed, there have been many methods proposed as well for extracting semantic relationships from a text corpus. These include the supervised method, distantly supervised method, bootstrap, OpenIE and others. The validity of these methods has been verified by experiments in different extraction systems. In this section, we briefly introduce these effective methods of RE.

2.1 Open IE

Open information extraction [4] is a domain-independent information extraction paradigm. Open IE systems are initialized with a few manually or automatically generated provided domain-independent extraction patterns, which are used to identify correct extractions from natural language. REVERB [10] uses simple syntactic and lexical constraints on binary relations expressed by verbs to match open information. ClausIE [7] separates the detection of "useful" pieces of information expressed in a sentence from their representation in terms of extractions. ReNoun [23] generalizes from a seed set to produce a set of extractions of noun attributes. Relation extraction is a special kind of information extraction so that Open IE technology has been sometimes used for relation extraction. Mapping information triples to relation triples can be accomplished in various ways, such as by manual rules [28], co-occurring relations in a large distantly labeled corpus [2], topic model [14] and clustering method [9].

In contrast, our approach does not require a large number of syntactic patterns corresponding to information, and we convert the dependency parse into a semantic dependency structure, multiple order semantic tree (MOS tree), through a small number of semantic dependency patterns, and obtain semantic information from the MOS tree.

2.2 Weakly supervised method

Weakly supervised methods to RE are attractive because they require markedly fewer manual training instances than supervised approaches. It uses the knowledge base or a small amount of labeled data to cover as many as possible valuable syntactic patterns of a relation.

The method is widely used in the automatic annotation of the knowledge base. This method is also called distant supervision [13, 16, 17, 18, 20, 22] and is found to have multi-label problems in applications. To deal with this problem, [20] proposes a novel approach to multi-instance multi-label learning, which jointly models all the instances of a pair of entities in text and all their labels using a graphical model with latent variables. Xiang et al. [22] model bias-dist and bias-reward and three noise-tolerant models to make use of the noisy training data. There is another method with the help of a few labeled data, called bootstrapping [12, 25, 27]. Zhang et al. [27] propose a kernel-based model, by which the most similar semantic shortest dependency patterns are selected to update seed patterns in each iteration of bootstrapping. On the basis of the weakly supervised method, Moro et al. [16] exploiting advanced comprehensive semantic knowledge resources can significantly improve extraction performance. Riedel et al. [18] propose the universal schemas to reduce dependence on training data. Moreover, Angeli et al. [2] apply a combination of distant and partial supervision to a relation extractor using a small number of carefully selected examples.

However, unlike those works that do the pre-learning for target relation types, the preparation for our work is only the understanding of relation types and entity types, which are easy for adding the new relation.

While weakly supervised learning has reduced the need for tagged data, its semantic drift and the need of labeled data for new relations require manual updating [33, 34]. Our approach does not need to learn the corpus since new relation labels only need the refinement targets without affecting the system.

2.3 Unsupervised method

Unsupervised relation extraction algorithms collect pairs of co-occurring entities as relation instances, extract features for instances and then apply unsupervised clustering techniques to find the major relations type.

Some researchers improve the representation of features [1, 6, 15, 19, 24]. De Lacalle and Lapata [6] present a new model which operates over tuples representing a syntactic relationship between two named entities and clusters tuples into underlying semantic relations by incorporating general domain knowledge which we encode as first-order logic rules. Moreover, there are some solutions to polysemy relations from the point of the algorithm. Min et al. [15] incorporate various knowledge sources to explicitly disambiguate polysemous relation phrases and groups synonymous ones. Wang et al. [21] propose a novel clustering algorithm which introduces co-clustering theory by the k-means clustering, not only clustering the entity pairs but also clustering the relation feature to make full use of the duality of the data. Rusu et al. [19] apply the method to extract events from the clustered complex news. Yu et al. [26] apply the multi-dimensional truth-finding to the unsupervised filling slot.

Although our method uses the techniques of clustering, it also uses a semantic dependency structure to filter the noisy information from sentences into a triple format which increases the precision of extraction. This is novel from the direct use of dependency parse.

3 Multiple order semantic pattern

3.1 Multiple order semantic

In addition to structured data, most relational semantics are contained in the unstructured text. Relational semantics, which express relations between entities, are organized into sentences by syntax and lexical rules with additional information. The semantic information in the knowledge base is expressed in the resource description framework (RDF). Its formal definition is as follows:

$$RDF = \langle Resource, Property, Propertyvalue \rangle$$

Each information, called *RDF* knowledge, consists of *Resource*, *Property* and *Propertyvalue*.

Resource is anything that holds a URI in the web.

Property is a resource with a name.

Propertyvalue is a specific value of a *Property*.

The combination of resources, attributes and attribute values in a natural statement forms a statement. In a statement, the *Resource* is the *Subject*, the *Property* is the *Predicate*, and the *Propertyvalue* is the *Object*. Here we also use these concepts to designate the semantic structure of sentences. According to syntactic and lexical features of a sentence, we define five basic relational semantic expression patterns of the sentences from unstructured text.

3.1.1 Basic Pattern

1. Simple pattern

$$SP = \langle S, P, O \rangle$$

The simple pattern, which is a direct pattern for a single relational semantic, is the most basic pattern of semantic expression. It consists of Subject(S), Predicate(P) and Object(O) and is the base of other patterns. As shown in Fig. 1a, the example sentence ("Google buys YouTube!") is a paradigm of the simple pattern for the relational semantic ⟨"Google," "buys," "YouTube"⟩. Moreover, this paradigm is not only limited to the "subject-verb-object" and "subject-copula-predicative" clause level structure but also applies to fragments of sentences such as noun phrases may be the simple pattern.

2. Compound pattern

$$CP = \langle SP, C, SP \rangle$$

The compound pattern is an expression pattern of a compound conjunction between two simple pattern semantics joined by a connective(C). In the compound sentences, there may be additional non-nominal clauses containing relational semantics besides the main clause. And each clause may contain parallel elements. Relational semantic

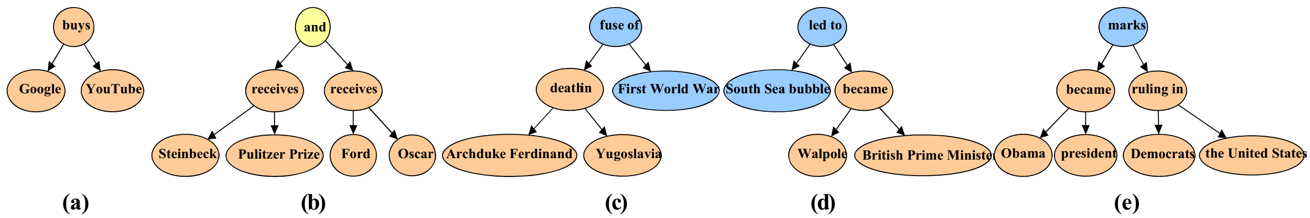


Fig. 1 Examples of five basic patterns

needs inference to obtain additional information, and they can make conjunct together. "Steinbeck receives a Pulitzer Prize, and Ford receives an Oscar." is an example sentence of compound pattern, shown in Fig. 1b.

3. Subject-nested pattern

$$SNP = \langle SP, P, O \rangle$$

The subject-nested pattern, in which the *Subject* of the main relational semantic is a nested relational semantic expressed by simple pattern, is a nested double-layer structure of two relational semantics. The subject-nested pattern may appear in complex subject sentences, such as subject clause, nominal phrase subject. "Archduke Ferdinand's death in Yugoslavia is the fuse of the First World War." is an example sentence of the subject-nested pattern, shown in Fig. 1c.

4. Object-nested pattern

$$ONP = \langle S, P, SP \rangle$$

Analogous to the subject-nested pattern, the object-nested pattern is also a nested double-layer structure of two relational semantic. The difference is that in the object-nested pattern, the *Object* of the surface semantic information itself is a binary semantic described by the simple pattern. It results in a nested semantic structure of the right child of the surface semantic structure. A sentence with a complex structure such as object clause, predicative clause and the nominal object/predicative is under the object-nested pattern. For example, the object-nested relational semantic in the sentence "South Sea bubble led to that Walpole became the first real British Prime Minister." is shown in Fig. 1d.

5. Full-nested pattern

$$FNP = \langle SP, P, SP \rangle$$

The full-nested pattern, as its name implies, is an expression pattern under which both the *Subject* and the *Object* contain nested relational semantic expressed by the simple pattern. "That Obama became president marks the Democrats' ruling in the United States" falls under the full-nested pattern as presented in Fig. 1e.

3.1.2 Hybrid pattern

In practice, the semantic structure of natural statements may be more complex. A single one of the above five patterns defined in the previous section cannot completely express a sentence's structure, and it is necessary to mix several patterns for full representation, which we call the "hybrid pattern." The hybrid pattern is defined as follows, it is a triple that connects the *Subject* part, and the *Object* part by a Connective(C) or Predicate(P) in which the *Subject* part may be a *Subject* or a nested hybrid pattern and the *Object* part may be an *Object* yet another nested hybrid pattern.

$$HP = \begin{cases} \langle HP, P/C, HP \rangle \\ SP|CP|SNP|ONP|FNP \end{cases}$$

In addition to the simple pattern, the other four basic patterns and hybrid patterns are all multilayered structures. According to the theory of higher mathematical logic, we call this multilayered semantic "Multiple Order Semantic." "Multiple Order" means that the multiple semantic information is nested or compound in a hierarchical structure, and each node contains a complete semantic. Restricted to the dependency features and surface features in the manual pattern set of the training corpus, the current methods can only extract some target semantic information from the multiple order semantic, even under the hybrid pattern. However, our goal is to do as much as possible to extract all semantic information in each natural statement.

3.2 Multiple order semantic tree

It is sometimes difficult and inefficient to extract semantic information directly from complex natural statements that contain the multiple order semantic. However, we can first obtain its multilayered semantic structure through the semantic pattern described in the previous section and then efficiently extract its complete semantic information. The primary goal of semantic information extraction is the construction of the multilayer semantic structure from natural statements. Therefore, we need to define a structure to describe the multiple order semantic of natural sentences. Since most meaningful semantic information of the multiple order semantic can be transformed into a binary relation, the binary tree is used here to define this multi-layered semantic structure.

Multiple order semantic tree (MOS tree) is a strict binary tree and is defined to store the multiple order semantic by six patterns described in the preceding section. The MOS tree is comprised of a set of nodes ($D_{\text{semanticnode}}$) and edges between them (R_{pattern}). Its formal definition is as follows:

$$T_{\text{MOSTree}} = \langle D_{\text{semanticnode}}, R_{\text{pattern}} \rangle$$

$D_{\text{semanticnode}}$ is a node that is a part of semantic information. In this paper, there are three types of nodes in MOS tree, namely relational phrase, relational subject/object and conjunction. These nodes are fragments of natural statements that contain semantic information.

R_{pattern} is a special part of the MOS tree, which is present on pairs. It is built on the previous six semantic patterns which combine into the above three kinds of nodes which make up the MOS tree.

An example of the MOS tree is shown in Fig. 2. Next, we introduce the three kinds of nodes in detail.

- Relational phrase

A relational phrase is a non-leaf node in the multiple order semantic tree which expresses the information in semantic. As a non-leaf node, the child of a relationship phrase may be any node: a relational subject/object, a conjunction, or a relational phrase. A relational phrase is a fragment of a natural state. It is not restricted to predicate verbs, and it may be a noun phrase, the copula (be), or preposition (in, of, etc.). The nodes "won," "won Nobel Prize for," "discovery of" and "born in" are all relational phrases of the sentence in Fig. 2.

- Relational subject/object

In the multiple order semantic tree, the relational subject/object is a leaf node that expresses the subject or object associated with the semantic. It acts like a child of a relational phrase in the multiple order semantic tree. It is usually composed of fragments in natural statements such as entities, adjectives, numbers, etc. As Fig. 2 shows, there are eleven relational subject/object nodes in the example, including "Nobel Prize," "Radium," "Polonium," "Madame Curie," "1867," "Warsaw." In particular, the semantic subtree "Madame Curie's discovery of Polonium" and "Madame Curie's discovery of Radium" can be used as the Relational Object of "won Nobel Prize for."

- Conjunction

The conjunction is another non-leaf node in the MOS tree that connects semantic subtrees. Thus, it is often the root node of a MOS tree of complex natural statements. Distinct from the relational phrase, the child of a conjunction can only be a relational phrase or another conjunction, representing semantic subtrees. According to the connection type between the two semantic subtrees, the conjunction is divided into the following two kinds:

Coordination (Node#CD) It means a coordination (and, but, or) between relational semantics pair in MOS tree. As shown in Fig. 2, three coordination conjunction nodes are linking the six relational semantics.

Complex (Node#CP) It means a complex connection between a compound relational semantics pair in MOS

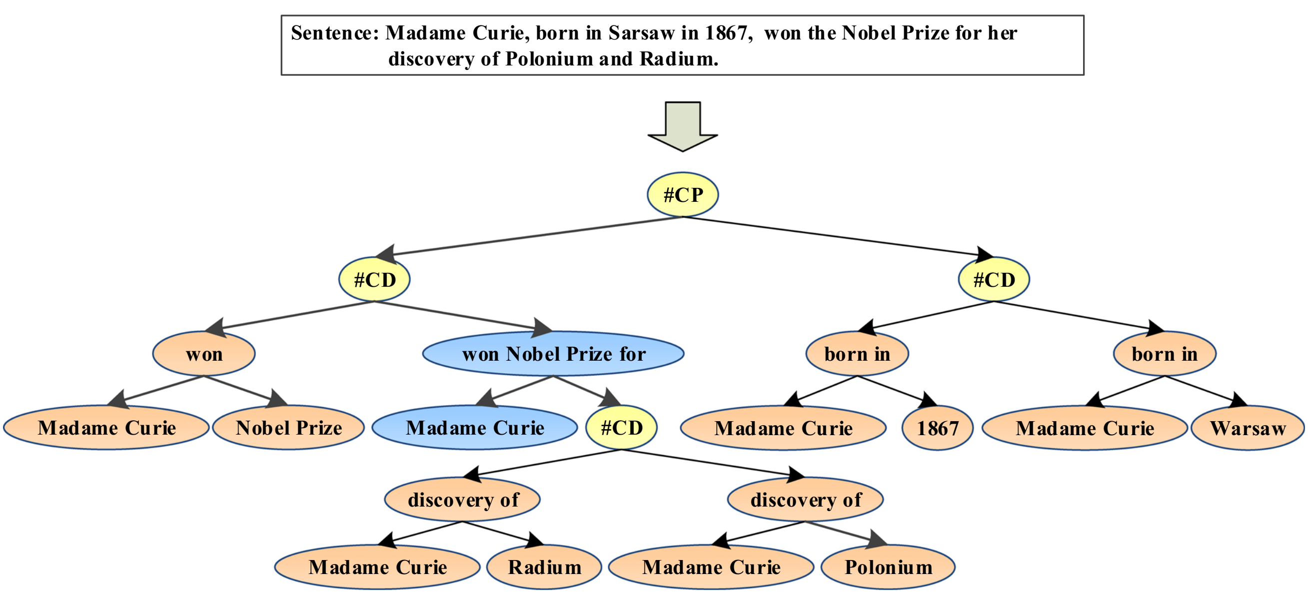


Fig. 2 An example of MOS tree

tree. It may appear in multiple compound relational semantics from a sentence. The complex conjunction nodes in Fig. 2 link the two semantic trees with multiple relational semantics.

With the semantic dependency structure of multiple order semantic tree, we analyze the multiple binary relations in natural statements and do not need to study each target relation. A variety of expression patterns for massive relationships can be covered as much as possible, and the method does not change by adding new relationships, which can solve the first problem. Also, the multiple order semantic tree can filter semantic information with binary relations and remove non-binary semantic information, unrelated entity pairs. and other noise information without semantic information. For example, the dependency path of an entity pair may pass through some relational phrases. But only one expresses the relation between the entity pair elements, while the other is the noise information in the analysis of the relation. The relational phrase of a MOS tree is directly connected a relational subject and object derived from the semantic analysis.

4 MOS parsing-based relation extraction

In this section, we describe the proposed MOSRE framework for extracting open-domain relations in unstructured text from the web and document. As shown in Fig. 3, this system consists of two modules: a multiple order semantic parser and a refinement processer.

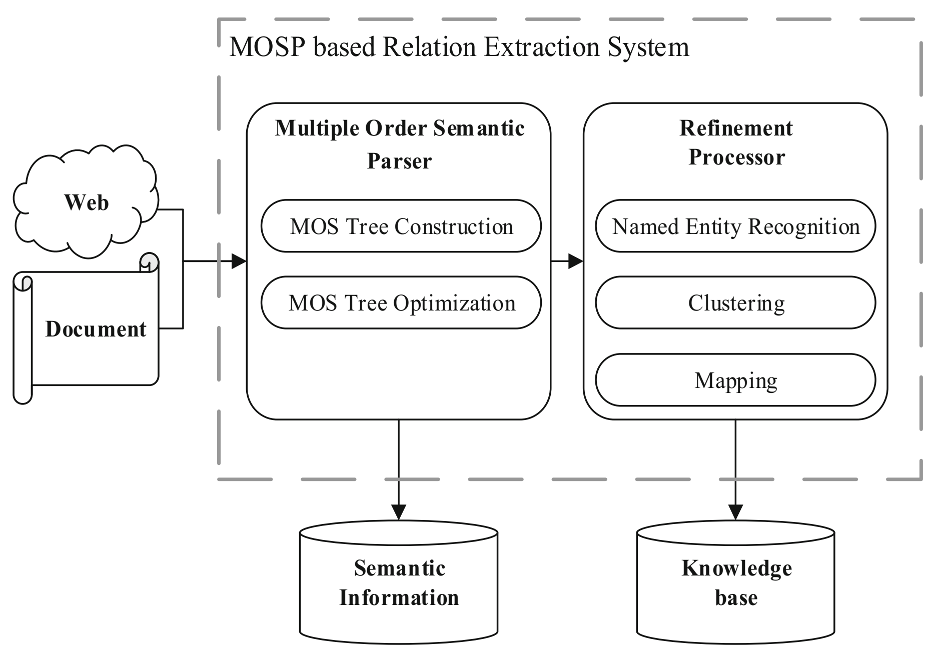
4.1 Multiple order semantic parser

The work of this module is to parse the semantic information in the unstructured text. The input is unstructured text from the web and document, and the output is structured semantic information in the form as $\langle S, P, O\rangle$. The multiple order semantic parser parses the semantic information by constructing the MOS tree of unstructured text. The method of generating a MOS tree, called "multiple order semantic parsing" (MOS parsing), is a process of scattering and restructuring as shown in Fig. 4. Firstly, sentences are preprocessed to remove the noise information. And then, sentences are processed to split into fragments with semantic association information. This step is completed by the NLP parsing tools. Next, these semantics are processed into a MOS tree by construction and optimization.

4.1.1 Construction

We will convert the dependency parsing into a semantic binary tree in this step. Our MOS parsing makes use of dependency parsing (DP) to reorganize the fragment to construct the MOS tree. The construction of a MOS tree from a relational phrase is described as MOST() in algorithm 1.

Fig. 3 Framework for MOSRE



Algorithm 1: MOS Tree construction MOST()
Input: Dependency Parsing results
Output: MOS Tree

```
While (find relational subject rs of RootKey)
    Lc = TreeNode(rs);
    If(RootKey.leftchild == null)
        RootKey.leftchild = Lc;
        MOST(RootKey.leftchild);
    Else
        Add a conjunction node as the parent of RooKey.ledtchild and Lc;
        MOST(Lc);
While (find relational object ro of RootKey)
    Rc = TreeNode(ro);
    If(RootKey.rightchild == null)
        RootKey.rightchild = Rc;
        MOST(RootKey.rightchild);
    Else
        Add a conjunction node as the parent of RooKey.rightchild and Rc;
        MOST(Rc);
```

In order to get the MOS tree, a root node is essential at the beginning of the parsing. The relational phrase of the main clause is selected as the initial root node because it is also the root of the dependency path. Starting from the initial root node, we first find the possible relational subject/object. The corresponding patterns are learned from many MOS tree examples and its dependency parsing in a corpus. The relational subject of a relational phrase may be the subject, possessive, a relational phrase of the nested clause and others. The relational object of a relational phrase may appear in some dependency structures such as the direct object, preposition structure, relational phrase of object clause. If the current root does not have a child node (subject, object or relational phrase), construction of the semantic tree is completed. Otherwise, the child node will be a new root node for recursive construction.

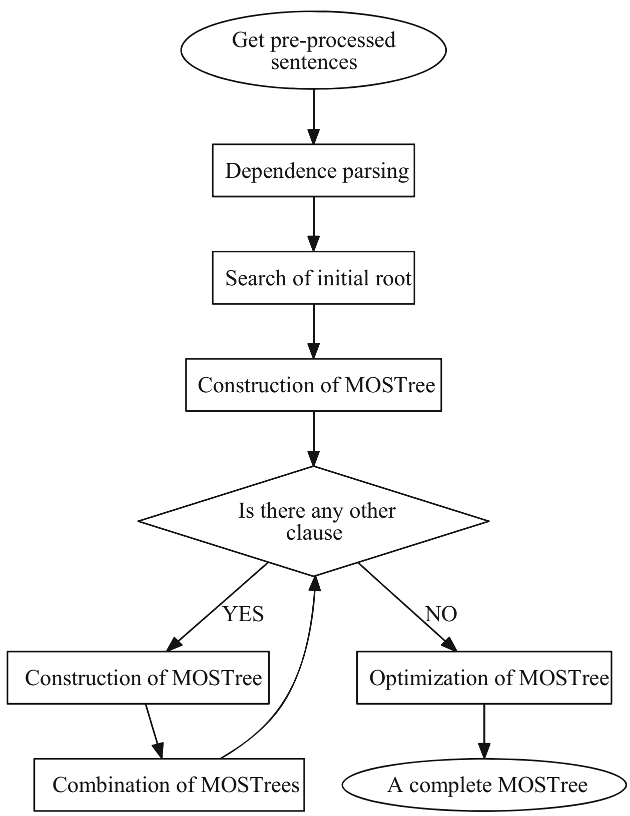


Fig. 4 Flowchart of MOS Parsing

For compound-complex sentences, there is more than one clause. After construction of a MOS tree from the initial root, the next step is to find additional clauses from the main clause (which is the origin of the MOS tree from the initial root). For each "brother," a conjunction node will be generated as a parent node to connect its root and brother. The brother node will as a root for recursive construction. After the completion of construction of each clause and conjunction, we get a semantic binary tree results.

4.1.2 Optimization

The semantic generating tree may contain single child subtree (only leftchild or only rightchild). Therefore, the following optimization work is to upgrade it into the strict binary tree.

$$T_{singlechild} = \begin{cases} T_{SPO}, T_{hasbinarysemantic} \\ Node, others \end{cases}$$

The specific work depends on whether the subtree has complete semantic information. As shown above, different solutions to these two cases are designed to adjust the structure of the subtree. If it has binary semantics, the subtree will be processed into a strict binary tree. Otherwise, the semantics of these subtrees are processed to merge into a semantic object node as a "LeafNode." Finally, a MOS tree will be successfully constructed, and each semantic will be obtained in the triple format by traversing the tree. Taking the sentence in Fig. 2 as an example, the generating MOS tree by this module is a four-layer structure. A total of seven statements of semantic information can be obtained from this MOS tree shown in Fig. 5.

4.2 Refinement processor

A refinement processor is a module that identifies the target relation from relational phrases. The input is the set of entity semantic information in the form $\langle S, P, O \rangle$. The output is the entity relation ⟨*Entity1, Relational label, Entity2*⟩. Polysemy and synonyms are both challenges in refinement. The same relationship might be expressed by different relational phrases in different sentences. For example, "purchase," "buy," "complete acquisition of" and "acquire" may express the same relation. However, the same relation phrases between different entity pairs may express different meanings. For the relation phrase "born in" in Fig. 4, the meaning between entity pair ⟨"Madame Curie," "Warsaw"⟩ is "birthplace," but the meaning between entity pair ⟨"Madame Curie," "1867"⟩ is "birthdate." This module uses the similarity computation to group the relation phrases into a cluster, which can then map to a known relation label. This process is made up of three steps: named entity recognition, clustering and mapping.

4.2.1 Named entity recognition

This component is used for identifying the entity to screen the semantic information to become an entity semantic information. There is a detailed list of the entity types in our system (titles, school, nationality, organization, person, date, country, city, province, prize, matter). This is due to the fine-grained difference of ours target relation. Some relations (such as "cities_of_residence" and "countries_of_residence") need detailed classes of content ("City" and "Country") rather than a general one ("Location"). Here we use the same method in Stanford NER to build the NER tools. It provides a general implementation of arbitrary order linear chain conditional random field sequence models by training sequence models on labeled data.

Also, non-entity information will be moved to get the relational phrases. Finally, entity relation triples are screened from the generated semantic triples and marked with content information in a format (like ⟨*Entity*1*, relational phrase, Entity*2⟩).

4.2.2 Clustering

In our system, we choose Markov clustering (MCL) to finish the task. MCL decomposes a word graph into small coherent pieces via simulation of random walks in the graph that eventually get trapped in dense regions, the resulting clusters. As reported in [8], MCL will effectively place each triple into exactly one cluster. It measures the semantic cohesiveness of a word's neighborhood as the curvature (also referred to as clustering coefficient of the word of the graph) and divides the word graph into classes of similar words by removing words of low curvature which connects several dispersed clusters. In detail, our algorithm is.

<Madame Curie, won, Nobel Prize>
<Madame Curie, discovery of, Radium>
<Madame Curie, discovery of, Polonium>
<Madame Curie, born in, 1867>
<Madame Curie, born in, Warsaw>
<Madame Curie, won Nobel Prize for, Madame Curie's discovery of Radium>
<Madame Curie, won Nobel Prize for, Madame Curie's discovery of Polonium>

Fig. 5 Statements containing semantic information in example sentences

Algorithm 2: Markov Clustering MCL()
Input: Graph G
Output: Clustered Graph G'

```
Add self-loop to G
Associated link graph G' = standardization (G)
Do {
    expansion(G')
    inflation(G')
} While (! check convergence)
```

The clustering work starts with the original graph G, which is the similarity matrix of candidate triples. Firstly, we add "self-loop" to expand the matrix. And then, the associated link graph G' is constructed from the original graph G by normalizing the graph. Next, MCL is applied to G'. This step is divided into two subroutines: expansion and inflation. The responsibility of the expansion is to connect different regions of the connection graph, while the inflation operation increases and decreases the current probability. These two steps are iterated until the matrix converges. It merges clusters whose overlap in information exceeds a certain threshold. Instead of clustering words by partitioning the original graph G, MCL clusters word contexts by partitioning the associated link graph of G'. The nodes in G' need to be based on contextual information and thus contain a nearly unambiguous meaning, where MCL is suitable for dividing the contexts into similarity classes.

The most essential and important steps are to calculate the similarity between triples. To better measure similarity, a novel similarity measure, denoted by $mix \sim (t_i, t_j)$, is designed to capture the degree of similarity between any two relation triples (t_i, t_j). The similarity is defined as:

$$mix \sim (t_i, t_j) = wn(t_i, t_j) \times con(t_i, t_j)$$

$wn(t_i, t_j)$ is the similarity of the relation triples in literal meaning to deal with the synonym problem. In this step, we use a semantic word similarity model constructed by combining latent semantic analysis (LSA)-based word similarity and WordNet knowledge to get a similarity score between relational phrases on the unit interval [11]. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph is constructed from a large piece of text and singular value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows. Values close to 1 represent very similar words, while values close to 0 represent very dissimilar words.

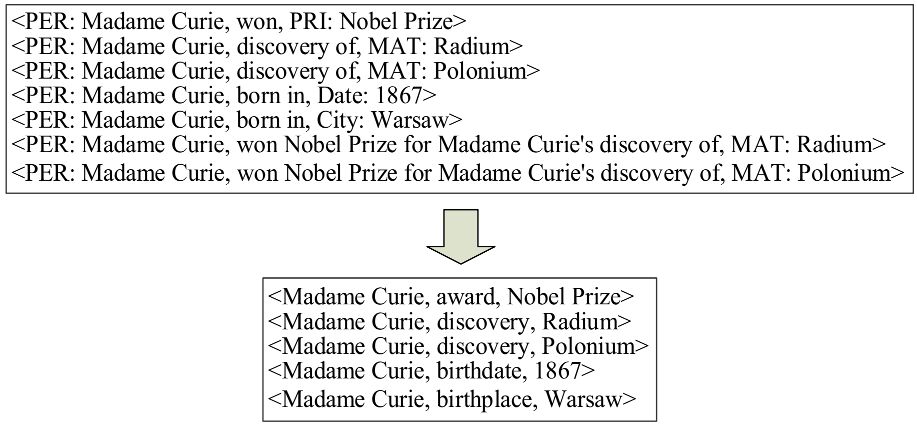


Fig. 6 Refinement result

Additionally, in order to eliminate polysemy, context semantic similarity ($con(t_i, t_j)$), which is based on named entity, is forced into the similarity measurement.

$$wn(x, y) = sim_{LSA}(x, y) + 0.5e^{-\alpha D(x,y)}$$

The similarity between word x and y by combining LSA similarity and WordNet relations is shown above. LSA word similarity relies on the distributional hypothesis that words occurring in the same contexts appear to have similar meanings. It applies singular value decomposition (SVD) to the word by word matrix and selects the 300 largest singular values and reduces the 29 K word vectors to 300 dimensions to perform SVD. Then, it defines a word's "significant senses" to cope with the problem of WordNet trivial senses for combining the WordNet knowledge with statistical word similarity measures. $D(x, y)$ is the minimum path distance between x and y in WordNet. According to previous research, $e^{-\alpha D(x,y)}$ is a very effective transformation of simple shortest path.

4.2.3 Mapping

The task of refinement is to unify the different forms of expression. The mapping work is done based on their similarity. First, every relation label is processed into the same format to facilitate the calculation of similarity. For example, the relational label "acquisition" applies to ⟨"*PER,*" "*birth,*" "*date*"⟩. Next, our system calculates the cluster center and the relation labels to get the highest similarity label for the entity relation. It is important to note here that the similarity higher than the threshold value can be successful in filtering out meaningless or unknown relation information from candidates. The semantic information in Fig. 5 is refined into several entity relations as shown in Fig. 6. The top of Fig. 6 shows the result of NER and the bottom shows the results output from the refinement processor. During the NER processing step, "1867" is recognized as an instance of "date," and "Warsaw" is recognized as an instance of "place." In the next step, the instances of "date" and "place" are mapped to "birthdate" and "birthplace" in knowledge base.

5 Experiments

In this section, we evaluate the practical performance of our proposed method. The experiment is divided into two parts. First, several parameters involved in the system are determined experimentally. Then, according to the optimal parameters, the test is carried out on the standard datasets. To avoid uncertainty, we run experiments on two test datasets, where the relations on one dataset are distinct while the relations on the other dataset are posing a greater challenge to semantic extraction.

5.1 Data

We evaluate the MOSRE-based relation extraction system on SENT500 and KBP datasets, respectively.

SENT500 The corpus contains 500 sentences from web documents, in which each sentence has at least one pair of entities. In fact, the SENT500 dataset includes 26 unique entity pairs, and each sentence has an identified entity pair in one of the four relations: "acquisition," "birthplace," "inventor," "win (award)." The four relations are invoked as the gold standard.

KBP The dataset is a version of MIML-RE annotation dataset provided by [3]. They use both the 2010 and 2013 KBP official document collections, as well as a July 2013 dump of Wikipedia as the text corpus to the values experimentally found in this section. Semantic in each natural statement parses information into a triple. There are 33,814 sentences annotated by the KBP relation labels (25 directional relations of person, 15 directional relations of organization and an additional "no relation" relation, shown in Table 1). Specifically, many sentences in this dataset are complex sentences.

5.2 Experiment result

5.2.1 System parameters

The refinement step involves the setting of two parameters: clustering parameters I and mapping threshold β. Here we use a testing dataset, which is built by randomly selecting 20 sentences for each relation type from the SENT500 and KBP, and perform experiments ten times.

Cluster parameter I in MCL determines the number of the matrices to influence in calculating the final granularity of clusters. Nakashole et al. [17] propose the calculation method of clustering scores in different granularity requirements. We measure the scores of different cluster parameters in the range 2–40. After ten iterations, $I = 20$ is selected as optimal.

For a given cluster parameter, we alter the threshold β in the range 0–1.0 to achieve high precision and recall. As shown in Fig. 7, the overall trends of three $F1$ curves of the average value of different times are similar: with the rise in threshold β, $F1$ value first has a rising stage and then begins to decline to the lowest point. By comparing the optimal values of each time shown in Table 2, we take the average of the ten experiments as the final value ($\beta = 0.07$). These triples clustered in a mixed way with relation labels to analyze clustering results. It is found that most of the triples, which have a similarity with relation labels over 0.2, are grouped with the relation label. Therefore, the expected value range of the threshold β is 0–0.2, mapping some extracted triples with low similarity to their respective relation labels. It is seen that the range of the experimental results is in the interval, which met our expectation. In the following experiments, these two parameters are assigned to the values experimentally found in this section.

5.2.2 SENT500

On SENT500, the proposed method (MOSRE) is compared against previous works on relation extraction to verify its effectiveness. We compare our method with following three extraction systems:

- *StatSnowBall* [24] proposes a statistical bootstrapping relation extraction using discriminative Markov logic networks and softens hard rules by learning their weights in a maximum likelihood estimate sense.
- *Hybrid* [4] uses a corpus-based characterization of binary relationships to learn a relation-independent extractor.
- *PROP* is a relation extraction method proposed by [5] that exploits the dyadic nature in semantic relations within a co-clustering framework.

We firstly measured the performance of our proposed method on the four relations of SENT500 and obtained the results as shown in Fig. 8. Each item in Fig. 8 consists of its relation type, the percentage of the dataset made up by this relation and the precision/recall value. We find that our method has a comparative performance with precision over 84%. In particular, the precision of acquisition and win/award are both above 90%. It is shown that the proposed method achieves a great performance on SENT500. Besides, in addition to relation "birthplace," the recall of other relations is higher than 75%, which is a great score. Except the relation "birthplace," the recall of other relations is higher than 75%, which is competitive with the state-of-the-art, because representations of this relationship may rely on symbols

For more insight into the performance of the proposed method, we report its performance of our proposed method compared with the three traditional relation extraction systems mentioned above. The results in Table 3 associated with the traditional methods are derived from their original publications. The $F1$ scores displayed are optimal among different situations.

From Table 3, we can see that the proposed method has the best macro-average precision and F-measure among all methods evaluated. The bold numbers are optimal results in each column in the table. The recall of our proposed method is higher than Hybrid and StatSnowBall. Although our system and PROP both use the clustering method to mapping the target relation, our system performed better than PROP in precision (over 15.7%) and $F1$ (over 3.6%). Our system has a recall rate exceeding that of hybrid and StatSnowBall, while marginally worse than that of PROP.

In order to better understanding of multiple order semantic relation extraction on this dataset, we apply a fine-grained decomposition of the experiment results. Table 4 presents the performance of our proposed method on total extractions and on four main surface patterns of extractions: VERB, VERB + PREP, NOUN + PREP and INFINITIVE.

In the first two surface patterns, VERB and VERB + PREP, the target relation is mostly shallow or surface

Table 1 Relations in KBP

People	Organization
per:title	org:top members/employees
per:stateorprovinces of residence	org:stateorprovince of headquarters
per:stateorprovince of birth	org:subsidiaries
per:spouse	org:shareholders
per:schools attended	org:parents
per:religion	org:member of
per:parents	org:founded by
per:origin	org:founded
per:employee of	org:country of headquarters

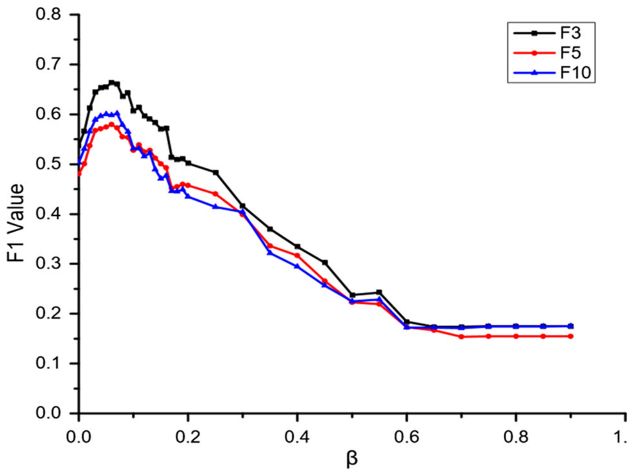


Fig. 7 $F1$ value curves with different β values

Table 2 β values in different experiments

Times	1	2	3	4	5
Optimal	0.06	0.09	0.06	0.05	0.05
Times	6	7	8	9	10
Optimal	0.07	0.07	0.08	0.08	0.07

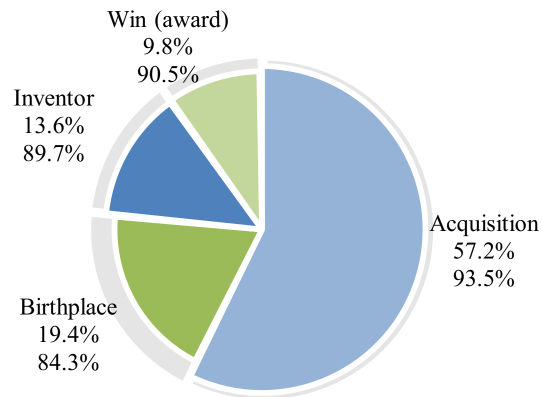
semantics. And in the other two, NOUN + PREP and INFINITIVE express the relation by the deep and nested structure of sentences, which reflects the difficulty of relation extraction from unstructured text. As shown in Table 4, the VERB structure relation phrase has achieved the best accuracy, recall and $F1$ values. Moreover, we can see that the result of our system on shallow semantics (VERB and VERB + PREP) is better than nested.

5.2.3 KBP

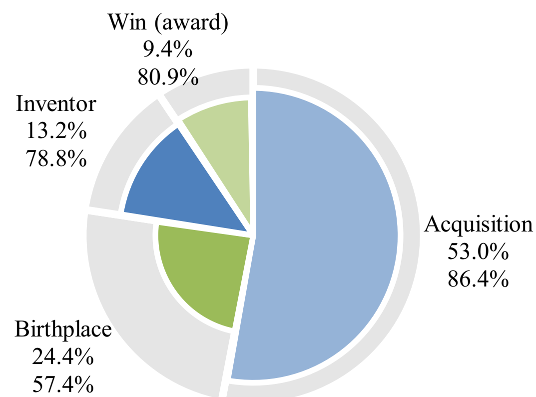
We used the same configuration on the KBP to do experiments again to evaluate performance. As a more complex dataset, KBP can better validate the performance on complex sentences.

The labeled data in KBP consist of 41 relations. In order to analyze, we divide it into two groups according to the subject: Person and Organization. The label in Fig. 9 is the same as in Fig. 8. As shown in Fig. 9, the precision on group "Organization" is higher than on group "Person", while the recall and $F1$ values on group "Person" are higher than on group "Organization."

Then we compare our proposed method proposed by Angeli et al. (a MIML-RE-based active learning method), which is known to achieve an excellent result on the TAC-KBP EF task. In the experiment of Angeli et al.'s system,



(a) Precision



(b) Recall

Fig. 8 Performances of four relations in SENT500. **a** Precision, **b** recall

the dataset is divided into two groups, 21,000 for training (about 70%) and 9000 for the test (about 30%). Then, we test our system under different situations to evaluate the effect of excluding each step in MOSRE. The specific configurations are as follows:

- *MOSRE-NMO* Extraction without multiple order semantic extracting. Extracting semantic relations follows a single semantic for each clause.
- *MOSRE-NCS* Similarity computation without context. The similarity of a relation phrase is used as the only similarity metric.
- *MOSRE-NC* Refinement without clustering. Direct mapping for each triple to target relations in refinement.

Table 5 shows the performance of the Angeli et al.'s system and results of our method under different configurations. The first line is the result of Angeli et al.'s system. And the following lines are the results of MOSRE-NMO, MOSRE-NCS and MOSRE-NC, respectively. Then, the last line is the performance of our proposed method with all

Table 3 The best performance on SENT500

	P	*R*	*F*1
Hybrid	79.2	56.9	66.2
StatSnowBall	79.8	73.3	76.4
PROP	75.2	**86.0**	80.2
MOSRE	**90.9**	77.8	**83.8**

steps included described in the last section. The bold numbers are optimal results in each column in the table.

The following observations are drawn from the statistical data presented in Table 5.

- Comparing the results of MOSRE-NMO, MOSRE-NCS, MOSRE-NC and MOSRE methods, we found the precision, recall and *F*1 value are increased step by step. That is to say that our proposed method is effective and reasonable.
- Compared with Angeli et al.'s system, the result shows that our proposed method gets higher precision and *F*1 score in the manual evaluation. The precision and *F*1 value are improved 20 and 5%, respectively.

Different from the high performance on SENT500, we have better results on KBP than other methods, but the *F*1 value is still lower. We first analyze the errors which arise in the results of our proposed method. Through a detailed view of errors in extracted triples and its original text, the error analysis on KBP is shown in Fig. 10. We consider that the likely reasons include:

- Incomplete extraction of semantic information: The present method for dependency parsing and our proposed MOS parsing cannot completely cover all the cases.
- Inaccurate similarity calculation. Relational phrases may be long and noisy or possibly words nonexistent in WordNet, causing an incorrect similarity calculation which leads to some errors in the refinement result step.
- Mapping errors: The mapping threshold obtained by the optimal *F*1 value resulted in certain mistakes during mapping.

In addition to a few errors caused by the first reason, numerous errors are produced due to the second and the third reasons during the refinement step. Based on further analyses, we consider these errors may be caused by our method's use of literal meaning to calculate similarity, which cannot fully express their latent relationships.

Furthermore, we performed further analysis to understand the reasons; our proposed method achieves a recall score of 0.4% lower than that MINL-RE as shown in Fig. 11. We found that relations under the following three cases have not been extracted. Precisely: 1) Unextracted Semantics: Some semantics are expressed through a number of compound semantics or hidden under another semantic. 2) Unidentified Semantics: After analysis of filtered out triples in the step of name entity recognition, there are some expected triples have not been recognized. 3) Unrefined Semantics: The relations labeled in some sentences do not correspond to their literal meaning and thus cannot be refined further.

6 Conclusion and future work

We defined a new pattern, named multiple order semantics, to express hierarchical, relational facts from complex structure sentences. Based upon this pattern, we proposed a novel method to extract semantic relations from unstructured text. Precisely, the proposed method obtains multiple semantic information by three steps of multiple order semantic parsing, named entity recognition and refinement. Our experimental results show that the new method has a significant improvement on relation extraction from sentences, especially for complex structure sentences. Hence, our proposed method not only solves the problem of multiple order semantic information extraction but also provides the additional advantage that the model can be improved by adding new relation labels without needing to require the model be retrained.

Although the experimental results show that our method achieves good performance on the dataset, there are also some aspects to improve. The first is about the multiple order semantic parser. There is some target relational information not directly resolved, which requires our system not only to improve the relevant rules for semantic information under special circumstances but also to cover semantic information on non-standard statements as much as possible. The components of the refinement processer also need improvement. Figure 11 shows that we should build a better corpus to retrain our model in the NER tools

Table 4 Fine-grained results on SENT500

	TOTAL	VERB	VERB + PREP	NOUN + PREP	INFINITIVE
P	90.88	97.50	90.00	88.00	93.10
R	77.80	94.10	79.30	62.90	81.80
*F*1	83.84	95.77	84.31	73.36	87.08

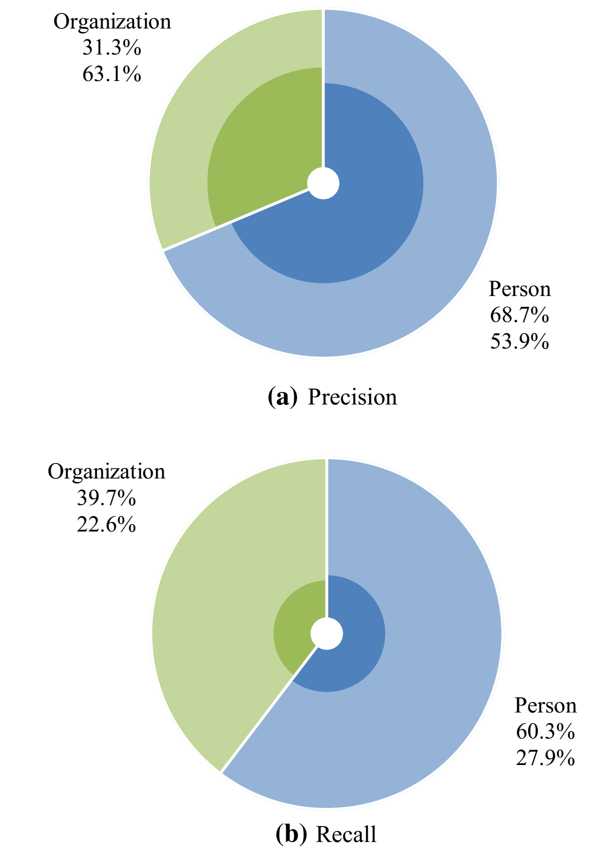


Fig. 9 Experimental results on KBP. **a** Precision, **b** recall

Table 5 The best performance on KBP

	P	*R*	*F*
Angeli	36.8	**26.2**	30.6
MOSRE-NMO	37.5	8.7	14.1
MOSRE-NCS	18.8	4.7	7.5
MOSRE-NC	35.1	15.7	21.7
MOSRE	**56.8**	25.8	**35.5**

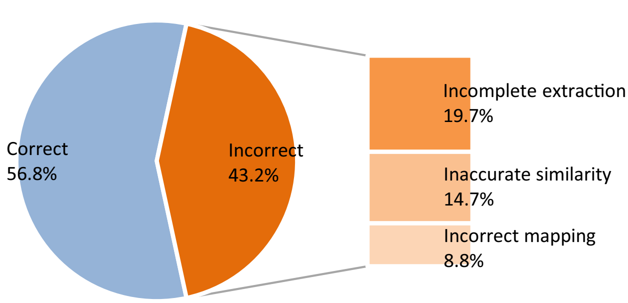


Fig. 10 Error analysis on KBP

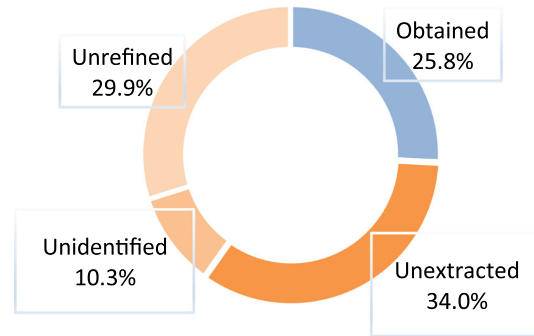


Fig. 11 Recall analysis on KBP

for identifying more entities. Then, the calculation should be improved to incorporate different kinds of semantic triples that do not exist in WordNet.

References

1. Akbik A, Visengeriyeva L, Herger P et al (2012) Unsupervised discovery of relations and discriminative extraction patterns. In: COLING, pp 17–32
2. Angeli G, Premkumar MJ, Manning CD (2015) Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd annual meeting of the association for computational linguistics (ACL 2015)
3. Angeli G, Tibshirani J, Wu J et al (2014) Combining distant and partial supervision for relation extraction. In: EMNLP, pp 1556–1567
4. Banko M, Cafarella MJ, Soderland S et al (2007) Open information extraction from the web. In: IJCAI, vol 7, pp 2670–2676
5. Banko M, Etzioni O (2008) The tradeoffs between open and traditional relation extraction. In: Proceedings of ACL-08: HLT, pp 28–36
6. De Lacalle OL, Lapata M (2013) Unsupervised relation extraction with general domain knowledge. In: EMNLP, pp 415–425
7. Del Corro L, Gemulla R (2013) Clausie: clause-based open information extraction. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 355–366
8. Dorow B, Widdows D, Ling K et al (2004) Using curvature and Markov clustering in graphs for lexical acquisition and word sense discrimination. arXiv preprint cond-mat/0403693
9. Dutta A, Meilicke C, Stuckenschmidt H (2015) Enriching structured knowledge with open information. In: Proceedings of the 24th international conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp 267–277
10. Fader A, Soderland S, Etzioni O (2011) Identifying relations for open information extraction. In: Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, pp 1535–1545
11. Han L, Kashyap AL, Finin T et al (2013) UMBC_EBIQUITY-CORE: semantic textual similarity systems. In: SEM@ NAACL-HLT, pp 44–52
12. Kong B, Xu RF, Wu DY (2015) Bootstrapping-based relation extraction in financial domain. In: 2015 International conference on machine learning and cybernetics (ICMLC). IEEE, vol 2, pp 897–903
13. Liu J, Rui W, Zhang L et al (2016) Social relation extraction with improved distant supervised and word embedding features. In:

2016 IEEE international conference on big data analysis (ICBDA). IEEE, pp 1–5
14. Melamud O, Berant J, Dagan I et al (2013) A two level model for context sensitive inference rules. In: ACL (1), pp 1331–1340
15. Min B, Shi S, Grishman R et al (2012) Ensemble semantics for large-scale unsupervised relation extraction. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics, pp 1027–1037
16. Moro A, Li H, Krause S et al (2013) Semantic rule filtering for web-scale relation extraction. In: International semantic web conference. Springer, Berlin, pp 347–362
17. Nakashole N, Weikum G, Suchanek F (2012) PATTY: a taxonomy of relational patterns with semantic types. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics, pp 1135–1145
18. Riedel S, Yao L, McCallum A et al (2013) Relation extraction with matrix factorization and universal schemas. In: HLT-NAACL, pp 74–84
19. Rusu D, Hodson J, Kimball A (2014) Unsupervised techniques for extracting and clustering complex events in news. In: ACL 2014, p 26
20. Surdeanu M, Tibshirani J, Nallapati R et al (2012) Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics, pp 455–465
21. Wang J, Jing Y, Teng Y et al (2012) A novel clustering algorithm for unsupervised relation extraction. In: 2012 Seventh international conference on digital information management (ICDIM). IEEE, pp 16–21
22. Xiang Y, Zhang Y, Wang X et al (2015) Bias modeling for distantly supervised relation extraction. Math Problems Eng. https://doi.org/10.1155/2015/969053
23. Yahya M, Whang S, Gupta R et al (2014) ReNoun: fact extraction for nominal attributes. In: EMNLP, pp 325–335
24. Yao L, Riedel S, McCallum A (2012) Unsupervised relation discovery with sense disambiguation. In: Proceedings of the 50th annual meeting of the association for computational linguistics: long papers—volume 1. Association for Computational Linguistics, pp 712–720
25. Ye F, Shi H, Wu S (2014) Research on pattern representation method in semi-supervised semantic relation extraction based on bootstrapping. In: 2014 Seventh international symposium on computational intelligence and design (ISCID). IEEE, vol 1, pp 568–572
26. Yu D, Huang H, Cassidy T et al (2014) The wisdom of minority: unsupervised slot filling validation based on multi-dimensional truth-finding. In: COLING, pp 1567–1578
27. Zhang C, Xu W, Gao S et al (2014) A bottom-up kernel of pattern learning for relation extraction. In: 2014 9th International symposium on Chinese spoken language processing (ISCSLP). IEEE, pp 609–613
28. Schmitz M, Bart R, Soderland S et al (2012) Open language learning for information extraction. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics, pp 523–534
29. Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp 697–706
30. Carlson A, Betteridge J, Kisiel B et al (2010) Toward an architecture for never-ending language learning. In: AAAI, vol 5, p 3
31. Auer S, Bizer C, Kobilarov G et al (2007) Dbpedia: a nucleus for a web of open data. In: The semantic web, pp 722–735
32. Ferrucci D, Brown E, Chu-Carroll J et al (2010) Building Watson: an overview of the DeepQA project. AI magazine 31(3):59–79
33. Zhou Q (2018) Multi-layer affective computing model based on emotional psychology. Electron Commer Res 18(1):109–124. https://doi.org/10.1007/s10660-017-9265-8
34. Zhou Q, Luo J (2015) Artificial neural network based grid computing of E-government scheduling for emergency management. Comput Syst Sci Eng 30(5):327–335