Novel Method for Mining Semantic Relationships for Entities in WIS

Yan ZHANG¹,², Qingzhong LI¹,*, Hui LI¹, Xinyan HUANG¹, Xiangwei WANG³

¹School of Computer Science and Technology, Shandong University, Jinan 250100, China
²School of Com. Sci. and Tech., Shandong Univ. of Finance and Economics, Jinan 250100, China
³State Grid Shandong Electric Power Company, Jinan 250100, China

Abstract

Extracting semantic relationships for entities in Web Integration Systems (WIS) is an important step for further analysis and decision-making tasks. For example, for a search query in WIS that contains a company name, we can not only return the descriptions of the company, but list companies that compete with it and also ones that cooperate with it. It is not rare that an entity pair has more than one semantic relationship. However, existing researches on relation extraction assume that one entity pair has only one semantic relationship. We propose a novel method to mine multi-semantic relationships for a giving entity in WIS. We first extract related entities and corresponding contexts from web texts, then propose a clustering algorithm to cluster the related entities into different subsets, where each subset represents a semantic relationship to the entity. We evaluate our method by comparing it with the state-of-the-art approach using real-world dataset generated by search engine. The results show that the proposed approach is efficient in mining multi-semantic relationships for the giving entity from WIS.

Keywords: Relation Extraction; Web Integration System; Relation Clustering

1 Introduction

Web Integration Systems (WIS) aim to integrate structural data from multiple Web sources and provide comprehensive structured information for advanced queries and further analysis. We have been working on web data integration in market intelligence [1-3]. Most entities, such as products, accessory parts of products, manufactures etc. in the integrated system, are inter-related. But many of the relationships are hard to find by querying the structured information residing in the WIS, while web pages contain additional information that indicates semantic relationships between pairs of entities. Mining related entities and relationships for a target entity within the WIS is meaningful for further analysis and decision-making. For example, for a search query in...
WIS that contains a company name, we can not only return the descriptions of the company, but list companies that compete with it and also ones that cooperate with it.

There are challenges to mine semantic relationships to a target entity at Web scale. First, there are multiple lexical patterns to express a single semantic relation. For example, aside from the pattern $X \text{ acquired } Y$, an acquisition between companies $X$ and $Y$ can be expressed using patterns such as $X \text{ bought } Y$, $Y \text{ is acquired by } X$, etc. Second, there might exist more than one semantic relation between a pair of entities. Moreover, the scale and heterogeneity of web text make it costly and impossible to do time-consuming deep language processing.

To solve the problem, we propose an unsupervised method to extract entities and mine semantic relations to the target entity. Given a text corpus, the proposed approach first extracts all mentions of entities that co-occur with the given entity in the same sentence and the corresponding context existing around the pair of entities. Then we filter out entities that don’t appear in the WIS. An efficient clustering algorithm is proposed to identify all subsets of entities that describe different semantic relationships to the target entity. It is noteworthy that when a pair of entities has multi-semantic relationships (the pair has more than one semantic relationship), our clustering algorithm can cluster it into different subsets.

The reminder of the paper is organized as follows. Section 2 discusses previous approaches in the literature. Section 3 presents an overview of our method and discusses the details. Section 4 presents experiments for verifying the effectiveness of our approach and Section 5 concludes the paper with giving directions for future work.

# 2 Related Works

The most related work to our research is relation extraction. Relation extraction has been promoted by the Message Understanding Conference and Automatic Content Extraction program [11]. The task has been traditionally studied as to extract predefined semantic relations between pairs of entities in text. That is, giving a sentence $S$ and a relation $R$, does $S$ assert $R$ between two entities in $S$. The supervised methods [4-6] require a set of human-tagged examples of the predefined relations. Culotta et al. [7] model the problem of relation extraction as a one of sequence labeling and used CRFs to identify the relations in a given document. Zhang Hongtao [4] studies the problems of extracting relations between biomedical entities in millions of biomedical research articles. Specifically, they perform relation extraction on text in which the topic of each document is known in advance. Then, for each entity pair found in a document, their goal is to predict the relation between that entity pair from a finite set of pre-defined relations. In our problem however, we do not know the relations that must be extracted beforehand. Moreover, the need for manually annotated training data by these supervised relation extraction systems makes it difficult to apply them to large-scale free text relation extraction task such as relation extraction from the web.

Besides traditional research on relation extraction, the other related research is the Open Information Extraction. Open IE is a domain independent information extraction paradigm and has been studied in both the natural language document corpus [8], and the Web [9] environment to extract relation tuples. Open IE can extract unknown relations from heterogeneous corpora. In this sense, Open IE is close to our proposed method. But our method differs from the Open IE methods in several respects. On one hand, Open IE systems require human selected features
to learn a good extractor, while the proposed method doesn’t need. On the other hand, Open IE uses deep linguistic parsing techniques to label training examples. In our method, we use cheaper and little linguistic processing and depend on efficient representation for contexts that the entity pairs co-occur and clustering algorithm to do relation classification. Danushka Bollegala et al. [9] proposes an unsupervised method to extract semantic relations between entities on the web. Their method goes further to label the extracted relations. But to the best of our knowledge, all the Open IE researches don’t consider that a pair of entities has multi-semantic relationships, which is common for many entity pairs.

3 Mining Semantic Relations

Before going into details of the proposed method, we define some notations related to the problem.

**Definition 1** A Named Entity (NE) is a set of labels that refer to the same real world entity. Since different sources may use different expressions to represent the same entity, we use a synonym set to represent one entity. Formally, $NE = e$, where $e = \{label_1, label_2\}$.

**Definition 2** A context is the textual features that occur around the related entity pair. It consists of the exact text before, in between and after the mentioned entities. Formally, if the related entities are $e_1$ and $e_2$, the context of the entity pair $CXT = \{t_b, e_1, t_n, e_2, t_a\}$ with $t_b, t_n$ and $t_a$ respectively standing for the text before, the text in the middle and the text after the entities.

**Definition 3** A relationship is consists of all entity pairs that have the similar semantic relation according to their contexts. Formally, a relationship $r = (\{<e_i, e_j>\}, label)$, where $<e_i, e_j>$ represents the related entity pairs that satisfy the relationship $r$, label donates the semantic of the relationship.

Based on the definitions above, this paper studies the problem of mining semantic relations for entities in a web integration system, which can be described as follows.

**Problem:** Giving a web corpus and a named entity $e$ from the Web Integration System, we mine semantic relationships to the giving entity from the web documents and the corresponding sets of entities related to each semantic relationship.

3.1 Method overview

The entire workflow is composed of three parts: data fetching, pre-processing and relation mining. In data fetching stage, we propose to use text snippets returned by a Web search engine as an approximation of the context of two entities. Snippets are brief summaries provided by most Web search engines and a snippet contains a window of text selected from a document that includes the queried keywords. Using snippets as contexts is computationally efficient because it avoids the need to download the source documents from the Web and the information in the snippet is enough to meet our need. And in pre-processing stage, we extract related entities and the corresponding contexts from the Web corpus. We run a part-of-speech (POS) tagger and annotate each sentence with POS tags (http://nlp.stanford.edu/software/Tagger.shtml). To detect potential entities in
sentences, we use a noun phrase chunking tool (http://chasen.org/taku/software/yamcha/) and extract noun phrase chunks containing at least one proper noun. After get the potential entities, we compare them with the named entity set to filter unconcerned entities. Then we extract context from the remained sentences that contain the target entity and the other entity. In order to locate the words that indicate semantic relationship between the pair of entities, we use the O-CRF proposed by [10] to extract real relational context from the sentences. In mining relations stage, we use the proposed clustering algorithm to mine relations based on the related entities and their contexts from the former stage. We construct the named entity set beforehand by tracing back the Web sources that used in the WIS to search for alternative forms of spelling for every named entity residing in the WIS.

3.2 Relation clustering

Assuming that the set of all extracted related entities is $E = \{e_i|i \in [1,n]\}$, and that the corresponding extracted contexts are $C = \{C_{ei}|e_i \in E\}$, which represents all the contexts that appear around the entity pair $e$ and $e_i$ in the given Web corpus. Then we propose an efficient clustering algorithm to identify the subset of entities that describes a particular semantic relation. The highlight of the algorithm is that we can cluster one entity into different subsets if it holds more than one semantic relationship with the target entity. We first sort all the related entities in ascending order of total frequency in that the least frequency entity holds least semantic relations with the target entity.

Algorithm 1 Relation Clustering

| input: related entities set $E = \{e_i|i \in [1,n]\}$, contexts set $C = \{C_{ei}|e_i \in E\}$ (1 ≤ $j$ ≤ $n$). |
| output: clustered entities $S_E = \{E_1, E_2, E_k\}$, corresponding sets of contexts to each entity cluster $S_C = \{C_{E1}, C_{E2}, \ldots, C_{Ek}\}$. |

1: SORT(E) // sort the set of entities E in the ascending order  
2: for each $e \in E$ SORT(E)  
3: $S_e \leftarrow \{\}, S_C \leftarrow \{\}$ // initialize both entity clusters and context clusters to empty set  
4: $e \leftarrow POP(E)$ // returns the first entity $e \in E$ and removes $e$ from E  
5: $S_E \leftarrow S_E \cup \{e\}$ //set the least co-occurrence entity as the first cluster  
6: $S_C \leftarrow S_C \cup \{C_e\}$ //set the corresponding contexts as the first context cluster  
7: while $E \neq \{\}$ do  
8: $e \leftarrow POP(E)$  
9: ASSIGN($e, C_e, S_E, S_C$)  
10: end while  
11: return $S_E, S_C$

The point of the algorithm is that we can cluster one entity into different subsets if it holds more than one semantic relationship with the target entity. For each entity $e$, we traverse each context $c$ in the set $C_e$ and compare if there are contexts semantic similarily based on edit distance and WordNet [12].

The description of the proposed clustering algorithm is presented in Algorithm 1. The algorithm takes as its input $E = \{e_i|i \in [1,n]\}$, which is the related entity set, and $C = \{C_{ei}|e_i \in E\}$ (1 ≤ $j$ ≤ $n$), which is the set of corresponding context set for each entity in $E$. The output of the clustering algorithm is the set of entities clusters $S_E$, and the corresponding context clusters, $S_C$. 

...
Function $\text{ASSIGIN}$ presented in Algorithm 2 describes the process of measuring the similarity between the corresponding context subsets $S_{ce}$ of entity $e$ with each cluster set $s_c$ in $S_C$ and put $e$ into all possible $s_e$ by the semantic similarity of its context set to clusters in $S_C$.

### Algorithm 2 Similarity Assign

1: function $\text{ASSIGIN}(e, C_e, S_E, S_C)$
2: clustered ← false
3: $t_C ← \text{null}$
4: while $C_e \neq \{\}$ do
5: $c ← \text{POP}(C_e)$
6: for each cluster $s_c ∈ S_C$ do
7: $\text{sim} ← \text{compare}(c, s_c)$
8: if $\text{sim} = 1$ then
9: $s_c ← s_c ⊕ c$
10: $s_e ← s_e ⊕ e$
11: clustered ← true
12: break
13: end if
14: end for
15: if clustered=false then
16: $t_C ← t_C \cup c$
17: end if
18: end while
19: if $t_C \neq \{\}$ then
20: $s_e ← s_e \cup c$
21: end if

The sorting operations in Algorithm 1 require $O(|E| \log |E|)$ complexity for all related entities. This sorting operation is required only once at the start. The while-loop starting from Line 7 terminates after $|E|$ iterations and the inside while-loop starting from Line 15 terminates after $|C_e| \sum |C_{ei}(e_i ∈ E)|$ at most. So the overall time complexity of Algorithm 1 is $O(|E|^2 \sum |C_{ei}(e_i ∈ E)|)$.

### 4 Experiment

We evaluate the proposed relation clustering method with the real-world dataset generated by a Web search engine. In the following parts of this section, we set up the dataset and evaluation criteria. Then we compare and analyze the results of the proposed method and the co-clustering algorithm proposed in [9].

#### 4.1 Dataset

We designate the dataset as the WebRE, which is composed of Web texts gathered from Web search engine by querying the entity “iPhone 5S”. To get related entities and the corresponding
contexts, we use Web search engine to generate related snippets. Since the results returned by search engine are versatile information related to the query keywords, we first filter snippets that don't contain two entities for that they don't indicate any semantic relations to the given entity. Further, we delete duplicate snippets returned by search engine. This exists because there are websites copying data from others and exactly same snippets should not be computed twice. After the preprocessing, we get 4466 different snippets for “iPhone 5S” as the dataset. From the 4466 snippets, we extract 118 different entities related to iPhone 5S after pre-processing.

4.2 Evaluation criteria

We use recall, precision and F-measure to evaluate the performance of our method in relation clustering. For dataset WebRE, since there is no existing ground truth, we choose 4 most frequent relations (relations that hold the most entities, which are ACCESSORY (utensils that used to protect or decorate iPhone 5S)), COMPETETOR (devices that compete with iPhone 5S), SELLER (sellers that provide iPhone 5S), CARRIER (mobile operators that support iPhone 5S), (the relation labels are manually selected) to the target entity iPhone 5S) and manually label the entities belong to the 4 relationships. We use the manually labeled clusters and entities belong to them as ground truth to evaluate our method. We suppose \( A \) is the number of entities that belong to the four clusters (for entities that belong to more than one cluster, we count all its appearances), \( B \) is the number of correctly clustered entities, and \( C \) is the number of wrongly clustered entities. Based on the definition of \( A \), \( B \) and \( C \), the definitions of recall, precision and F-measure we use as follows:

\[
recall = \frac{B}{A}, \quad \text{precision} = \frac{B}{B + C}, \quad F1 = \frac{2 \times recall \times precision}{recall + precision}
\]

4.3 Experimental results and analysis

Table 1 shows the manually labeled clusters and the numbers of related entities belong to them. We use the manually labeled clusters and entities as ground truth to evaluate the efficiency of the proposed method.

<table>
<thead>
<tr>
<th>ACCESSORY</th>
<th>OPPONENT</th>
<th>SELLER</th>
<th>CARRIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>25</td>
<td>18</td>
<td>16</td>
</tr>
</tbody>
</table>

From manually labeling, we get that entities belong to relationships ACCESSORY and OPPONENT are single-semantic related to the target entity, which means that entities belong to these two subsets don't appear in other subsets; and some entities belong to SELLER and CARRIER are multi-semantic to the target entity, like China Mobile, it is the CARRIER for iPhone 5S, and at the same time, it is a SELLER of iPhone 5S.

In order to test the accuracy and scalability of our method, we compare our algorithm with the co-clustering algorithm proposed in [9] and we refer to it as CO-algorithm, while the proposed algorithm as PRO-algorithm. The CO-algorithm uses cosine similarity to measure the similarity of two entity pairs. After delicate calculating and proving, they choose the threshold to be 0.05
to judge whether two entity pairs are semantic similar or not. We use the same threshold when implementing their algorithm.

Fig. 1 demonstrates the recall, precision and F-measure of the two algorithms. Notice that we label four relations, but the picture only shows two of them. It is because that the CO-algorithm doesn’t distinguish \textit{SELLER} and \textit{CARRIER}, which should be separated because they represent different semantic relationship. The results show that the proposed algorithm and CO-algorithm are close in precision, recall and F-measure in clustering entity pairs that have single-semantic relationships. Table 2 shows the precision, recall and F-measure for relationship \textit{CARRIER} and \textit{SELLER} of the proposed algorithm. The recall of the proposed method is relatively low for the reason that when we compute the similarity of context to context sets of existing clusters, we don’t consider the semantics, which results in one or more clusters should be merged into one. But given the result clusters and contexts related to them, it is easy for user to identify which clusters are semantic similar and can be viewed as one.

![Table 2: Results of proposed algorithm in \textit{CARRIER} and \textit{SELLER}](image)

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{CARRIER}</td>
<td>90.91%</td>
<td>66.67%</td>
<td>76.92%</td>
</tr>
<tr>
<td>\textit{SELLER}</td>
<td>92.31%</td>
<td>54.55%</td>
<td>68.57%</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, a method to mine related entities and semantic relations for a target entity in WIS is proposed. Compared to other researches in relation extraction and clustering, the proposed algorithm can cluster one entity into different cluster when it has more than one semantic relationship to the target entity. Experimental results show that the proposed approach is effective in mining semantic relationships for entities in Web Integration Systems. The future work of our research in this area is to incorporate versatile web documents besides search engine results to enhance the recall of the mined relations and incorporating semantics of context when compare the similarity of contexts will enhance precision and it is our future direction in this area.
Acknowledgement

Supported by the Science and Technology Development Plan Project of Shandong Province (2014GGX101019), the Natural Science Foundation of China (61303005), the Shandong Distinguished Middle-aged and Young Scientist Encouragement and Reward Foundation (BS2012DX015), the Technology Project of State Grid (No. 2012WK515) and the Fundamental Research Funds of Shandong University (2014JC025).

References


