The Open Cybernetics & Systemics Journal, **2015**, *9*, **799-806**

Semantic Relationships Extraction for Entities in WIS

Zhang Yan[1,*] and Zhang Rui[2]

[1]*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, 250014, P.R. China*

[2]*Shandong Provincial Institute of Electronic Products Supervision & Inspection, Jinan, 250014, P.R. China*

Abstract: Web Integration System (WIS) provides abundant structured information about entities in a domain. Inter-relationships for entities in WIS are valuable for further analysis and decision-making. It is not rare that an entity pair has more than one semantic relationship. However, existing researches on relation extraction ignore this situation and they assume that one entity pair has only one semantic relationship. This paper focuses on mining *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. Finally we adjust the result clusters by merging semantic similar clusters together. The highlight of the algorithm is that we cluster the entities based on every single context instead of the overall contexts related. 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, named entity, 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.

In this paper, we concentrate on mining semantic relationships from external web documents to a target entity within the WIS. That is, with the target entity e, we need to find entities related to it and cluster them into different sets according to the semantics of relationships, which is as shown in Fig. (**1**). In the figure, there are three semantic relationships related to entity e, labeled as r1, r2 and r3 respectively. The entities in the ellipses are the corresponding entities with the specified relationship r_i to e. It is noteworthy that e_1 and e_3 in the figure have *multi-semantic relationships* to e.

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 acquired Y, an acquisition between companies X and Y can be expressed using patterns such as X bought Y, Y is acquired by X, etc. Second, there might be more than one semantic relation between a pair of entities. For example, two companies might have OPPONENT and SUPPLIER relations at the same time, like APPLE and SUMSANG (SUMSANG provides screens for APPLE products and at the same time SUMSANG products compete with APPLE products). The relation mining system must distinguish the different relationships that one entity holds to the target entity.

To solve the problem, we propose a novel method to extract entities and 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. A 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. Finally we adjust the result clusters by merging semantic similar clusters together. We test the proposed approach in real-world dataset generated by search engine and compare it with state-of-the-art approach. The results show that the proposed approach is

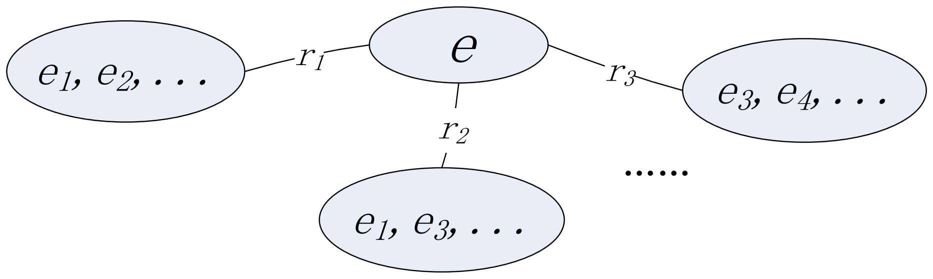


Fig. (1). Clustered entities related to target entity e.

close with state-of-the-art approach in clustering single-semantic relationship entity pairs and have good precision in clustering multi-semantic relationship ones.

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 [4]. 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 [5-7] require a set of human-tagged examples of the predefined relations. Culotta *et al.* [8] 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 [5] 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 [9], and the Web [10] 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.* [10] 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 not rare for many entity pairs.

Since our goal is to mine semantic relationships for entities in Web Integration System of market intelligence field, different from all the above work, which mainly concentrates on entity pairs, we have a target entity and try to mine semantic relations related to it. And more importantly, the above work in relation extraction and semantic measuring for different entity pairs don't consider the situation that there are more than one semantic relationship between a pair of entities, which is common in real-world and many application domains.

3. SEMANTIC RELATIONS CLUSTERING

We first define some notations used in this paper and then define the problem to solve. Based on this, we describe the workflow of our solution to the problem, followed by details on the steps involved in the procession.

3.1. Problem Definition

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 $e = \{label_1, label_2 \ldots\}$. For example, we use the set {*iPhone 5s, IP5s, Apple iPhone5s*…} to represent the cell phone *iPhone 5s* produced by *Apple*.

Definition 2: The *Named Entity Set (NES)* is a repository that holds named entities. Formally, the $NES = \{e_1, e_2\ldots\ldots\}$, in which e_i (i = 1, 2 …) is a named entity as defined above. In this paper the *NES* is composed of entities from *WIS*. Since the *WIS* has integrated data from different web sources, we have recorded the variants of entities during the process of integrating. This helps us to construct the named entity dictionary. We use the *NES* to filter out entities that don't reside in the *WIS*.

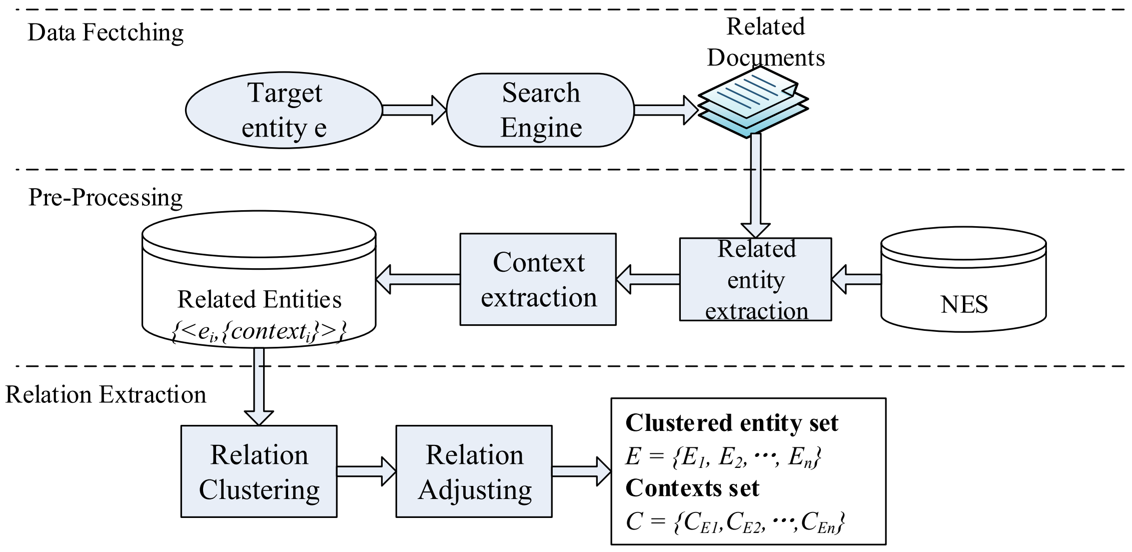


Fig. (2). Approach workflow.

Definition 3: 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 $CXT = \{t_b, e_1, t_{in}, e_2, t_a\}$ with t_b, t_{in} and t_a respectively standing for the *text before*, the *text in the middle* and the *text after* the entities. For example, for entities *Samsung* and *Apple*, the context for the sentence "*Samsung gains as Apple display supplier*" is *{"", e_1, "gains as", e_2, "display supplier"}*.

Definition 4: 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. For example, the given sentence "*Samsung gains as Apple display supplier*" donate a relationship r = *(<Samsung, Apple>, supplier)*. It is noteworthy that we don't distinguish the semantic details of a relationship. That is, in the above example we don't distinguish whether *Samsung supplies* displays or other things to *Apple*. We just cluster all entities that have "*supply*" relations with the target entity together.

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 named entity *e* from the Web Integration System and a collection of documents (web search results in the work described in this paper), we extract entities related to the given entity and mine semantic relationships between them in a fully unsupervised way, that is, we cluster the related entities into different subsets based on the contexts the entity pairs co-occur.

3.2. Workflow of the Approach

In this paper, we concentrate on mining semantic relations for the entities reside in the WIS. That is, giving an entity that exists in the WIS, we mine entities that have certain relationships with it by fetching and analyzing external information. The workflow of our approach to solve the problem stated in last section is shown in Fig. (**2**).

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 pro-processing stage, we extract related entities and the corresponding contexts from the Web corpus. We recognize named entities by referring to the named entity set, which was set up by entities from WIS. 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.

In the data fetching stage, we get the snippets returned by search engine as Web corpus, which are sentences. Based on the Web corpus, in the stage of pre-processing, we run a part-of-speech (POS) tagger and annotate each sentence with POS tags [11]. To detect potential entities in sentences, we use a noun phrase chunking tool [12] and extract noun phrase chunks containing at least one proper noun. Note that all the sentences contain the target entity since they are results of querying words represent the entity. 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 [13] to extract real relational context from the sentences.

3.3 Relation Pre-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. We propose the clustering algorithm to identify the subset of entities that describes a particular semantic relation to the target entity. First we sort the related entities in ascending order of total frequency in that the least frequency entity holds least semantic relations with the target entity. For each related entity e and the corresponding contexts c_e, the algorithm proceeds in three steps:

1. Separate the contexts in c_e into different set S_{ec} based on edit distance and word similarity in WordNet [14].
2. For each set s_{ec} in S_{ec}, compute the similarity of s_{ec} with each existing clusters s_c. Merge s_{ec} with s_c when the similarity is greater than threshold and e to the corresponding entity cluster s_e.
3. For each set s_{ec} that the similarity is less than threshold for all existing s_c, create a new cluster s_c' for context set s_{ec} and $\{e\}$ as the new corresponding entity cluster.

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 similarity based on edit distance and WordNet [14]. In step 2, when we compute the similarity between set s_{ec} and s_c, we compare each pair of contexts from both s_{ec} and s_c and take the maximum similarity value as the similarity for set s_{ec} and s_c. We don't eliminate entity e in the entity set until all the contexts in C_e are handled, which ensures that e can appear in different entity clusters when the contexts related to it indicate different semantics.

The description of the 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 $C=\{C_{ei}| ei\in E\}$ $(1<=j<=n)$, 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. In line 9 of Algorithm 1, we cluster the contexts based on edit distance and Wordnet. Function ASSIGN 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. The compare function in line 18 compares contexts of entity e with cluster S_c by comparing each single context in both sets and using the average similarity as the result.

Algorithm 1: Relation Clustering

```
Input: related entities set E=$\{e_i| i\in[1,n]\}$, contexts set C=$\{C_{ei}| ei\in E\}$ $(1<=j<=n)$.
Output: clustered entities $S_E=\{E_1, E_2,...E_k\}$, corresponding sets of contexts to each entity cluster $S_C=\{C_{E1},C_{E2},...,C_{EK}\}$.
1:  SORT(E)  // sort $E$ in ascending order of frequency
2:  for each $e\in E$
        SORT($C_e$)  // sort $C_e$ in descending order of frequency
3:  $S_E\leftarrow\{\}$, $S_C\leftarrow\{\}$
4:  $e\leftarrow$POP(E)  // first entity $e\in E$ and removes $e$ from $E$
5:  $S_E \leftarrow S_E\cup\{e\}$  //set the {e} as the first cluster
6:  $S_C\leftarrow S_C\cup\{ C_e \}$  //set {$C_e$} as the first context cluster
7:  while $E\neq\{\}$ do
8:    $e\leftarrow$POP(E)
9:    $S_{ce} \leftarrow$ cluster($C_e$)
10:   ASSIGN(e, $S_{ce}$, $S_E$, $S_C$)
11: end while
12: return $S_E$, $S_C$
13: function ASSIGN(e, $S_{ce}$, $S_E$, $S_C$)
14:   clustered$_{Se}\leftarrow$false  //initialize all context subset of
                                     // entity $e$ as not clustered
15:   while $S_{ce}\neq\{\}$
16:     $c\leftarrow$POP($S_{ce}$)
17:     for each cluster $s_c\in S_C$
18:         sim $\leftarrow$ compare($s_{ce}$, $s_c$)
19:       if sim>$\theta$ then
20:           $s_c\leftarrow s_c\oplus s_{ce}$
21:           $s_e\leftarrow s_e\oplus e$
22:           clusered$_{sce}\leftarrow$true
23:           break
24:       endif
25:     endfor
26:   endwhile
27:   for all clustered$_{se}$ = false
28:     $S_C= S_C\cup s_e$
29:     $S_E= S_E\cup\{e\}$
30:   endfor
```

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 17 in function ASSIGN terminates after $|C_e|\sum|C_{ei}(e_i\in E)|$ at most. So the overall time complexity of the Algorithm is $O(|E|^2|\sum|C_{ei}(e_i\in E)|)$.

3.4. Relation Clusters Adjusting

Due to the diversity of web texts, there are contexts that are not similar either in WordNet or edit distance but they have the same semantics. For example, "Google assimilates YouTube" and "finally Google actually bought YouTube" indicates the same relation ACQUISITION (between two companies, where one company is acquired by the other), but *assimilate* and *buy* are not similar both in edit distance and WordNet. In order to merge such context clusters together, we adjust the clusters based on distributional hypothesis [15], which follows that if two context clusters are distributed similarly over a set of related entities, then these context clusters must be semantic similar. To do that, we represent the related entities and corresponding context clusters in a matrix in which the rows correspond to entities, and columns correspond to context clusters. The A_{ij} element of the data matrix is either 1 or 0, where 1 denotes that entity e_j is in the entity cluster corresponding to cluster context c_i. The column vectors can be considered as defining the distribution of a context cluster over the space spanned by the related entities. The merging process works in the following steps:

1. For each pair of context clusters, compute the cosine similarity between their vectors ci and c_j.
2. If the similarity is greater than threshold, merge the context clusters and the corresponding entity clusters. Replace the context clusters with the new one in the matrix and update the corresponding column vector.

Repeat step 1 until the matrix doesn't change.

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 [10].

4.1. Dataset

In order to test the efficiency and scalability of the proposed method in real-world relation extraction and clustering tasks, we use Web search engine to generate related web corpus by querying the target named entity. 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 as stated in 3.2. 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. We also filter out named entities that don't exist in our Named Entity Dictionary as stated in section 3.2. 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 preprocessing as stated in section 3.2.

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:

$$recall = \frac{B}{A} \tag{1}$$

$$precision = \frac{B}{B+C} \tag{2}$$

$$F1 = \frac{2 \times recall \times precision}{recall + precision} \tag{3}$$

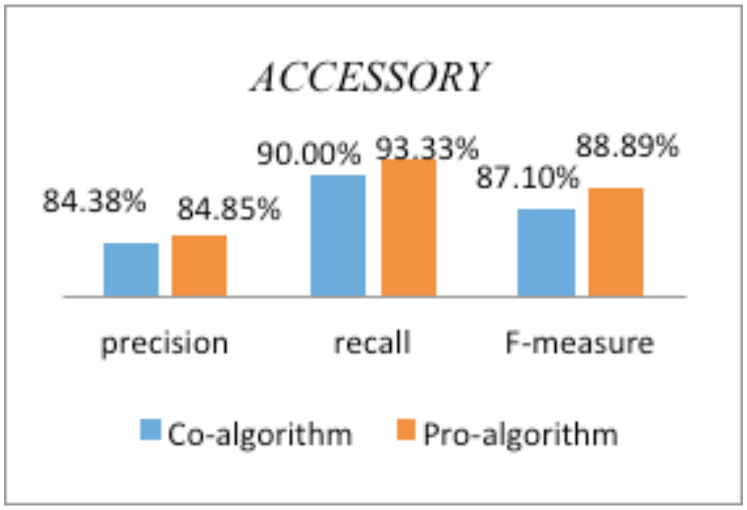
4.3. Experimental Results and Analysis

Since there is no existing ground truth, we choose four relations that hold the most related entities to the target entity and manually label the entities belong to them. 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 1. Relation clusters and numbers of entities in each cluster.

Accessory	Opponent	Seller	Carrier
42	25	18	16

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



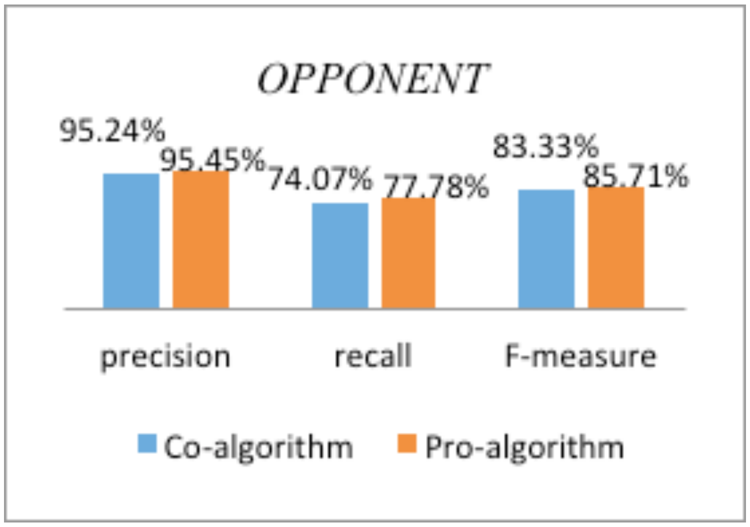


Fig. (3). Comparison results for *single-semantic* entity pairs.

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 [10] 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. And in the PRO-algorithm, we set the similarity threshold to be 0.6 in the pre-clustering and 0.8 in the cluster adjust step.

Table **2** shows the numbers of clusters generated by the two algorithms. From the table, we can see that the proposed method produce more clusters than CO-algorithm. It is because in the proposed method, we evaluate the similarity of each context of an entity pair to existing clusters to decide which clusters it belongs to, while the CO-algorithm takes all the contexts that related to an entity pair as a vector to compute similarity with existing clusters.

Table 2. Clusters in dataset WebRE.

	#clusters
CO-algorithm	17
PRO-algorithm	25

Fig. (**3**). 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 *SELLER* and *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 **3** shows the precision, recall and F-measure for relationship *CARRIER* and *SELLER* of the proposed method. The recall of the proposed method is relatively low for the reason that the sentences indicate *CARRIER* and *SELLER* relations to the *iPhone 5s* are relatively rare and there are less repeated entities appearing in different sentences, which results in that two or more clusters should have been merged into one.

Table 3. Results of proposed algorithm in *CARRIER* and *SELLER*.

	Precision	Recall	F-measure
CARRIER	90.91%	66.67%	76.92%
SELLER	92.31%	54.55%	68.57%

CONCLUSION

In this paper, a method to mine related entities and semantic relations for a target entity in WIS is proposed. We first fetch related web documents related to the target entity, and extract related entities and corresponding contexts that indicate semantic relationships to the entity. Then we propose an efficient clustering algorithm to cluster the related entities into different subsets by their corresponding contexts, where each subset represents a semantic relation to the given entity. Later we adjust the clusters by merging semantic similar context clusters based on distributional hypothesis to get more precise results. 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 enclose external named entities into the named entity dictionary besides those from the WIS to discover more potential relations for target entity. Furthermore, how to find a representative label for each relation cluster is our future work in this area.

In the early stages of the ceramic product design, the designer's requirements are often vague, the design may just remain in style with a brief description. Designers are generally based on their own experience, or according to the requirements of customers, from the case library to extract one or several programs, and then complete the selection and modification of the design with the clients. In this case, there is conflict between the designers of the program and the users are not satisfied with the existing program or part of the program, and also want their full participation in the scheme comments. Faced with these requirements of the customers in the design process, designers want to able to give quick feedback, timely conversion needs quick practicable sketch program, and modification in real time based on customer feedback.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Supported by the National Key Technologies R&D Program (2012BAH54F02), Shandong Province Independent Innovation Major Special Project (2013CXC30201), the Natural Science Foundation of China (61303005), the Shandong Distinguished Middle-aged and Young Scientist Encouragement and Reward Foundation (BS2012DX015), and the Fundamental Research Funds of Shandong University (2014JC025).

REFERENCES

[1] Z. Yan, Q. Li, S. Zhang, Z. Peng, Y. Dong, Y. Ding, Y. Zhang, and X. Xu, "MI-WDIS: Web data integration system for market intelligence", In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, 2010, pp. 1957-1958.

[2] Y. Dong, Q. Li, Y. Zheng, X. Xu, and Y. Zhang, "Semantic annotation of web object using constrained conditional random fields". In: *Proceedings of the 11th International Conference on Web-Age Information Management*, JiuZhaiGou, 2010, pp. 28-39.

[3] Y. Dong, Q. Li, Y. Ding, and Z. Peng, "A query interface matching approach based on extended evidence theory for deep web," *Journal of Computer Science and Technology*, vol. 25, pp. 1-11, Mar, 2010.

[4] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ace) program-tasks, data, and evaluation", In: *Proceedings of the Language Resources and Evaluation Conference*, European Language Resources Association, 2004.

[5] Z. Tao, "*Research on Automated Biomedical Relation Extraction from Bio-Literature*", Ph.D. thesis, TsingHua Universtity, 2012.

[6] G. Zhou, M. Zhang, D. H. Ji, and Q. Zhu. "Tree kernel-based relation extraction with context-sensitive structured parse tree information", In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, 2007, pp. 728-736.

[7] C. Giuliano, A. Lavelli, and L. Romano, "Exploiting shallow linguistic information for relation extraction from biomedical literature", In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics,* Trento, 2006, pp. 401-408.

[8] A. Culotta, A. McCallum, and J. Betz, "Integrating probabilistic extraction models and data mining to discover relations and patterns in text", In: *Proceedings of NAACL'06*, New York, 2006, pp. 296-303

[9] Y. Shinyama, and S. Sekine, "Preemptive information extraction using unrestricted relation discovery", In: *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics,* New York, 2006, pp. 304-311

[10] D.T. Bollegala, Y. Matsuo, and M. Ishizuka, "Relational duality: Unsupervised extraction of semantic relations between entities on the web", In: *Proceedings of the 19th International Conference on World Wide Web,* Raleigh, 2010, pp. 151-160.

[11] http://nlp.stanford.edu/software/tagger.shtml, [Online Resource], Accessed on March 1st, 2014.

[12] http://chasen.org/~taku/software/yahcha/, [Online Resource], Accessed on March 1st, 2014.

[13] M. Banko, and O. Etzioni. "The tradeoffs between traditional and open relation extraction", In: *Proceedings of ACL'08,* Columbus, 2008, pp. 28-36

[14] M.A. Finlayson, "Java libraries for accessing the Princeton Wordnet: comparison and evaluation," In: *Proceedings of the 7th Global Wordnet Conference.* Tartu, 2014, pp. 78-85.

[15] Z. Harris, "Distributional structure," *Word*, vol. 10, pp. 146-162, 1954.

Received: December 16, 2014 Revised: March 23, 2015 Accepted: June 31, 2015