A Hybrid Framework for Semantic Relation Extraction over Enterprise Data

Wei Shen, CCCE & CS, Nankai University, Tianjin, China

Jianyong Wang, Department of Computer Science and Technology, Tsinghua University, Beijing, China & Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou, China

Ping Luo, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Min Wang, Visa Inc., Foster City, CA, USA

ABSTRACT

Relation extraction from the Web data has attracted a lot of attention in recent years. However, little work has been done when it comes to relation extraction from the enterprise data regardless of the urgent needs to such work in real applications (e.g., E-discovery). One distinct characteristic of the enterprise data (in comparison with the Web data) is its low redundancy. Previous work on relation extraction from the Web data largely relies on the data's high redundancy level and thus cannot be applied to the enterprise data effectively. This paper proposes an unsupervised hybrid framework called REACTOR. REACTOR combines a statistical method, classification, and clustering to identify various types of relations among entities appearing in the enterprise data automatically. Furthermore, the authors explore to apply pronominal anaphora resolution to extract more relations expressed across multiple sentences. They evaluate REACTOR over a real-world enterprise data set from HP that contains over three million pages and the experimental results show the effectiveness of REACTOR.

Keywords *Anaphora Resolution, Enterprise Data, Information Extraction, Relation Extraction, Relation Tagging*

1. INTRODUCTION

Relation extraction is the process of discovering the relationship among two or more entities from a given unstructured data set. It is an important research area not only for information retrieval (Salton & McGill, 1986) but also for Web mining and knowledge base population (Shen, Wang, Luo, & Wang, 2012). The huge amount of valuable information contained in the unstructured text is recorded and transmitted every day in the text form. Turning such information into the understandable and usable form is of high significance and has a lot of real applications.

DOI: 10.4018/IJSWIS.2015070101

Traditional relation extraction processes usually require significant human effort: they need predefined relation names and hand-tagged examples of each named relation as input (Kambhatla, 2004 ; Zelenko, Aone, & Richardella, 2003 ; Giuliano, Lavelli, & Romano, 2006 ; Zhou, Zhang, Ji, & Zhu, 2007 ; Surdeanu & Ciaramita, 2007). Weakly supervised systems for relation extraction such as the bootstrapping systems require much less human involvements, but still require a small set of domain-specific seed instances or seed patterns that have a big impact on the system performance. Furthermore, the seed selection process requires substantial domain knowledge and is usually time consuming (Agichtein & Gravano, 2000 ; Zhu, Nie, Liu, Zhang, & Wen, 2009 ; Brin, 1998 ; Etzioni et al., 2005). Open IE is proposed as a new relation extraction paradigm that can identify various types of relations without predefinition. The goal of open IE systems is to gather a large set of relation facts that can be used for question answering (Banko, Cafarella, Soderl, Broadhead, & Etzioni, 2007 ; Banko & Etzioni, 2008 ; Etzioni et al., 2005 ; Shinyama & Sekine, 2006). Despite that, most relation extraction systems constrain the search for binary relations that are asserted within a single sentence (i.e., single-sentential relations) (Agichtein & Gravano, 2000 ; Zelenko et al., 2003 ; Brin, 1998 ; Zhu et al., 2009 ; Zhou et al., 2007 ; Hasegawa, Sekine, & Grishman, 2004), while relations between two entities can also be expressed across multiple sentences (i.e., inter-sentential relations). The analysis in Swampillai and Stevenson (2010) shows that inter-sentential relations constitute 28.5% and 9.4% of the total number of relations in MUC6 data set (Grishman & Sundheim, 1996) and ACE03 data set respectively. This places upper bounds on the recall of relation extraction systems that just consider single-sentential relations.

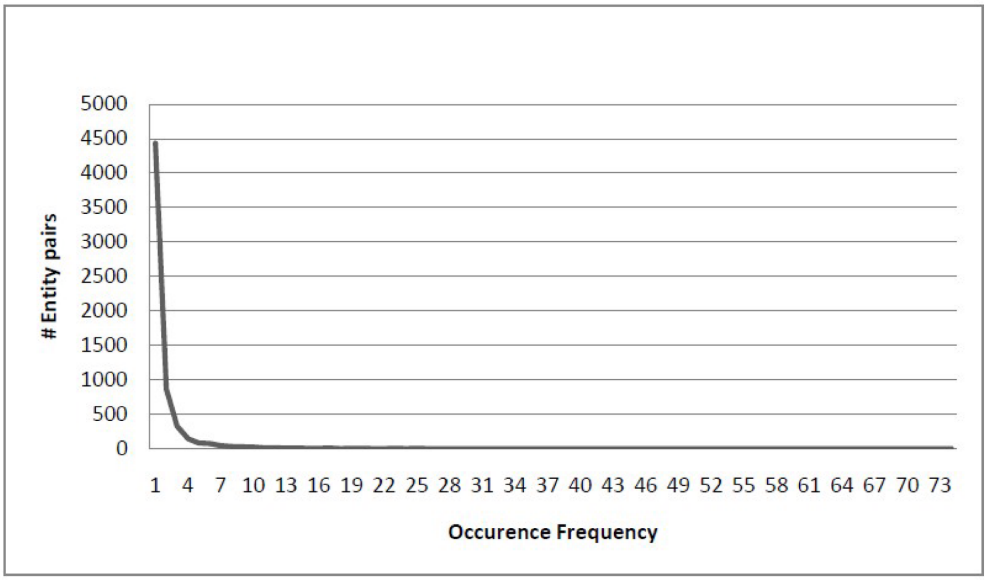
While most work on relation extraction focuses on the Web data, the amount of the enterprise data (including e-mails, internal Web pages, word processing files, and databases) has grown significantly during the past several years for all companies. Numerous real-world entities such as people, organizations, and products are contained in the enterprise data and these entities are connected by various types of relations. To make use of such rich information, it is desirable to build an entity relationship graph that can support efficient retrieval of entities and their relations. A key application of the entity relationship graph is in E-discovery, the process of collecting, preparing, reviewing and producing evidence in the form of Electronically Stored Information (ESI) during litigation (Crowley & Harris, 2007). In this process, lawyers need to find all the people and ESI that are relevant to a legal matter. For example, when a company is alleged to have infringed a patent related to a product, this company is required to disclose all the relevant information. The first question is which employees are closely related to this product. Furthermore, it will be more useful if we could provide their specific roles to this product, such as product manager, product support, or sales manager. To answer these questions, semantic relation extraction from the enterprise data is an essential step.

However, the existing techniques on relation extraction cannot be applied to the enterprise data directly due to the differences in the data characteristics: the enterprise data has much lower redundancy than the Web data. Figure 1 shows the distribution for the occurrence frequency of entity pairs for the PEOPLE-ORGANIZATION (PEO-ORG) domain in the enterprise data set used in our experiments. It shows that more than 90% of the entity pairs occur less than four times, about two thirds of the entity pairs only occur once in the entire data set and the average occurrence frequency of all the entity pairs is 1.96. In this paper, the occurrence of an entity pair means that the entities of that entity pair co-occur within the same sentence. Most existing techniques rely on the high redundancy nature of the Web data for an abundant supply of related entities to achieve reasonable recall. The recall will fall dramatically when applying such techniques to the low-redundancy enterprise data. Considering the sentence "… Bob, technology consultant for Software Division …" which just appears once in the data set, a good algorithm

should be able to extract the following relation: "Bob" (PEOPLE) is a "technology consultant" of "Software Division" (ORGANIZATION). However, the existing techniques can hardly discover it since they consider the relation only appearing once is unreliable. On the other hand, some other characteristics of the enterprise data could be leveraged for more effective relation extraction. For example, the enterprise data is less noisy than the Web data, and we usually have some known knowledge or databases within an enterprise that can be leveraged to support the entity recognition process. Therefore, we could exploit the existing useful information to minimize the human involvement and improve the performance of relation extraction on the enterprise data.

In this paper, we propose a novel unsupervised hybrid framework called REACTOR. It uses a statistical method in conjunction with the classification and clustering techniques to extract semantic relations and can label the extracted relations with representative tags over the enterprise data. It also applies the pronominal anaphora resolution techniques to extract inter-sentential relations. Specifically, given an enterprise data set where entities of interest have been identified already, REACTOR first adopts a statistical method to extract a set of representative entity pairs that contain both positive and negative examples for the classifier. Then we extract some features from the positive and negative examples to train the classifier that is in turn used to classify all the other entity pairs each of which appears in the same sentence as related or not. For each entity pair classified as related, a context vector consisting of the words from all its occurring sentences is generated, and a clustering algorithm is used to identify the semantic relations of entity pairs. Furthermore, to describe the semantic relations for the entity pairs in each cluster, REACTOR employs a closed frequent sequence pattern mining algorithm to extract some representative tags. To extract inter-sentential relations, we apply an anaphora resolution algorithm to the original documents and get the substitution text where pronominal references are substituted by the noun phrases they refer to. Accordingly, we transform inter-sentential relations expressed by the pronominal anaphora to single-sentential relations that can be processed by REACTOR. Subsequently, REACTOR uses the methods introduced above to process the

Figure 1. The distribution for the occurrence frequency of entity pairs (in PEO-ORG domain)



substitution text to extract the single-sentential and inter-sentential relations together. Note that a very preliminary version of the paper has been published as a poster in WWW'11 conference (Shen, Wang, Luo, Wang, & Yao, 2011). In this paper, we make further enhancements, and give a complete and in-depth description of our proposed REACTOR approach.

The main contributions of this paper are summarized as follows:

- We present REACTOR, a hybrid framework that can effectively extract semantic relations over the low-redundancy enterprise data. Most previous work on relation extraction is for the high-redundancy Web data.
- REACTOR is an unsupervised framework that requires minimal human involvement. It employs a statistical method to automatically generate the training data for the classifier.
- REACTOR can extract inter-sentential relations to significantly boost the recall of the system and the experimental results reveal that information referenced pronominally is very important to inter-sentential relation extraction.
- REACTOR can label each extracted relation with tags that describe the semantic relation accurately. It applies a closed frequent sequence pattern mining algorithm to extract the representative tags.
- We extensively evaluate REACTOR over a real-world enterprise data set that contains over three million pages. The experimental results show that REACTOR can achieve significantly higher precision and recall compared with the baseline method.

The rest of the paper is organized as follows. Section 2 discusses related work and Section 3 introduces the REACTOR framework. Specifically, Section 3.1 gives an overview and Section 3.2 describes how to extract the representative entity pairs. Section 3.3 presents a classifier that is used to detect related entity pairs. Section 3.4 describes how to extract the semantic relations using a clustering algorithm. Relation tagging is introduced in Section 3.5. Section 3.6 introduces the extraction of inter-sentential relations. Section 4 presents our experimental results and Section 5 draws conclusions.

2. RELATED WORK AND DISCUSSION

Relation extraction was first introduced in the Message Understanding Conference (MUC) (Grishman & Sundheim, 1996), and the Automatic Content Extraction (ACE) program promoted relation extraction as a task of Relation Detection and Characterization (RDC) in 2001, which was renamed to Relation Detection and Recognition (RDR) in the ACE 2004 evaluation.

Following these tasks, many supervised machine learning approaches were proposed such as maximum entropy models (Kambhatla, 2004), kernel methods (Zelenko et al., 2003 ; Giuliano et al., 2006 ; Zhou et al., 2007), Perceptrons (Surdeanu & Ciaramita, 2007) and hidden Markov models (Freitag & Mccallum, 1999 ; Skounakis, Craven, & Ray, 2003). These supervised methods need manually annotated training data to learn an extractor, which makes them difficult to be applied to large-scale relation extraction tasks like relation discovery over the enterprise data, since it is expensive and time consuming to obtain the human-labeled examples. Moreover, these methods usually extract a set of rules from the human tagged training data. The performance of the extracted rules will be very poor when they are applied to data with a different style. Consequently, we have to spend a great deal of time and effort to prepare a set of human tagged examples for each targeting style data when we apply them to the enterprise data that has diverse text styles and genres.

There are also some previous works that adopted weakly supervised learning approaches such as the bootstrapping systems (Agichtein & Gravano, 2000 ; Brin, 1998 ; Zhu et al., 2009 ; Etzioni et al., 2005). These approaches significantly reduce manual labor needed for relation extraction by only needing a small set of seed examples or seed extraction patterns. Beginning with these seeds, bootstrapping methods iteratively discover new extraction patterns and new instances. However, the selection of the seeds requires substantial expertise because the performance of bootstrapping systems heavily depends on the initial seed examples or seed patterns provided to them. It is also unclear how the initial seeds or patterns should be selected and how many seeds are needed, which confuses the non-expert users. Additionally, nontrivial manual effort is also required when shifting to a new relation extraction task since this method demands a set of hand-crafted seeds per relation to launch the training process. What is more, for the bootstrapping systems relations have to be specified in advance for the preparation of the initial seeds. In our setting, however, it is impossible to know the targeting relations beforehand in the enterprise data.

Open Information Extraction (Open IE) was firstly introduced in Banko et al. (2007) as a novel domain-independent relation extraction paradigm that works well on huge and diverse Web corpus. It eliminates the drawbacks of the traditional information extraction paradigm that relies on lots of human involvement in the form of manually tagged training data or hand-craft seed examples. Open IE has been studied in both the Web environment (Banko et al., 2007 ; Banko & Etzioni, 2008 ; Etzioni et al., 2005) and natural language document corpus (Shinyama & Sekine, 2006). Although these Open IE systems are promising and can be suitably applied to extract unknown relations from large scale heterogeneous corpora such as the Web corpus, they have some unsatisfactory aspects in comparison with REACTOR. First, the Open IE systems can just label the entity pairs as "trustworthy" or not and are unable to give users more descriptions about the extracted relations. REACTOR can go a further step which can identify the extracted relations with informative lexical descriptions that are very useful and important for extracting unknown relations. Second, although the Open IE systems are self-supervised, they still need a set of human-selected generic, domain independent patterns to create a set of extraction rules. While in REACTOR, we use a statistical method to select training examples for the classifier automatically. In addition, all Open IE systems rely on the high redundancy of the Web for an abundant supply of simple sentences that are relatively easy to process. When we come to the enterprise application where the redundancy is much lower than the Web data, this assumption is violated so that the recall of the Open IE system will fall drastically.

There are also some other completely unsupervised approaches for relation extraction (Hassan, Hassan, & Emam, 2006; Hasegawa et al., 2004). The method proposed in Hassan et al. (2006) is to extract patterns from *n*-gram language model and use an iterative procedure based on graph mutual reinforcement to identify highly confident patterns. In the approach of Hasegawa et al. (2004), clustering techniques are used for unsupervised relation extraction. Context vectors for entity pairs are composed of all words appearing between the entities, and they are clustered using cosine distance. Each generated cluster contains the entity pairs with the same relation type. Overall, these unsupervised methods all depend on the high redundancy level of the large corpora and have the assumption that useful relations will be mentioned frequently. They also assume that the relations mentioned once or twice are not likely to be important. Whereas in our setting of the enterprise application, most relations are mentioned just once or twice due to the data characteristic of low redundancy, accordingly, these unsupervised approaches are not suitable to be applied to the enterprise data.

3. THE REACTOR FRAMEWORK

3.1. Overview

In this subsection, we give you a brief introduction to the proposed REACTOR framework. Different modules will be explained in detail in the following subsections.

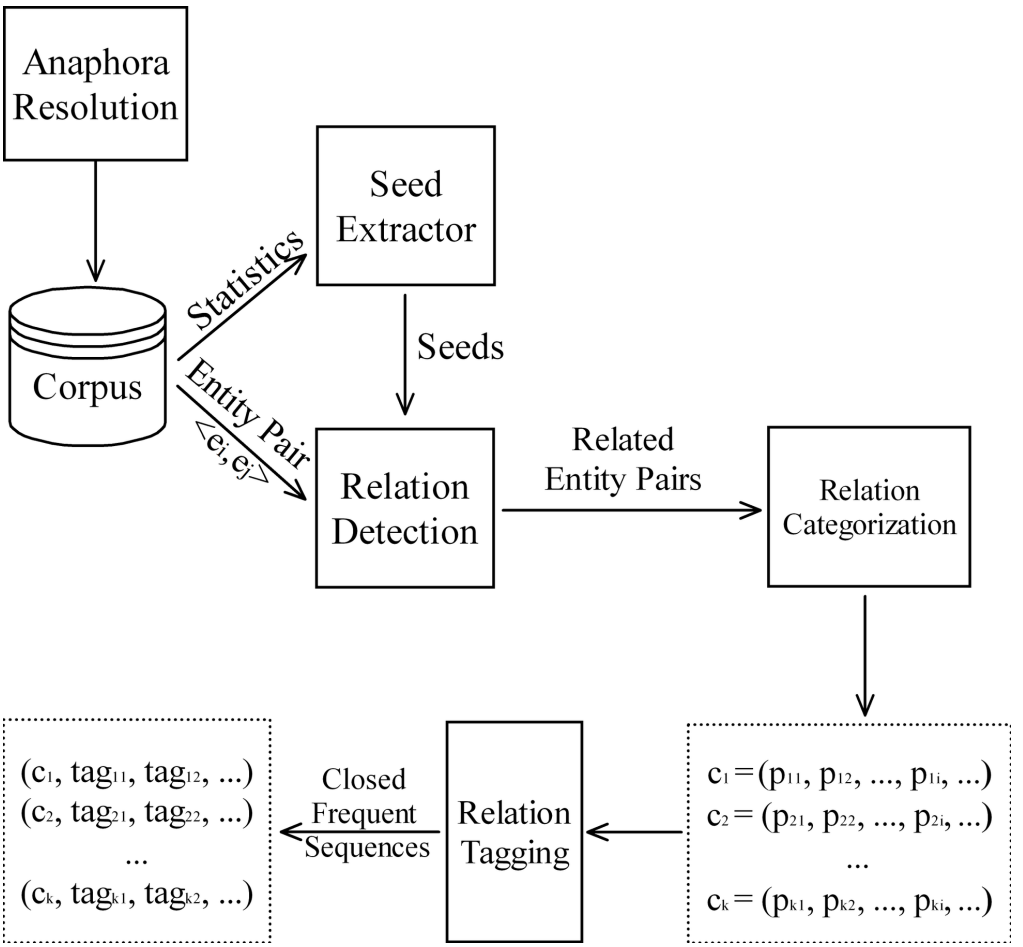
Given a text corpus, the goal of REACTOR is to extract all semantic relations between any two types of entities. We assume that entities of the two corresponding types in this corpus, T_m and T_n, are previously detected like many other relation extraction systems (Agichtein & Gravano, 2000 ; Zhu et al., 2009 ; Hasegawa et al., 2004) and moreover, the disambiguation process of these entities has been completed. Therefore, each detected entity in the corpus has an identifier that corresponds to a unique real-world entity. As the types and the number of the semantic relations possibly valid between two entities in a pair are unknown, our system aims to extract all related entity pairs with the types of T_m and T_n in the corpus, and select some representative tags to describe the semantic relation for each entity pair. Figure 2 depicts the architecture of REACTOR.

Generally speaking, REACTOR has five modules including Seed Extractor, Relation Detection, Relation Categorization, Relation Tagging, and Anaphora Resolution. The Seed Extractor uses statistics to extract a set of representative entity pairs containing both positive and negative seed examples to train the classifier in the Relation Detection module. Specifically, the Seed Extractor applies a form of *pointwise mutual information* (PMI) between two entities e_i and e_j to assess the probability whether a relationship exists between these two entities. The Relation Detection module classifies each entity pair $< e_i, e_j >$ with the targeting types T_m and T_n occurring within the same sentence as related or not. The sentence boundaries in each document are found using the OpenNLP toolkits[1] which can perform sentence detection. Then for each entity pair classified as related, a context vector consisting of words formed from all its occurring sentences can be generated. The third module Relation Categorization employs the hierarchical clustering algorithm to produce several clusters (e.g., $c_1, c_2, \ldots, c_k$) and in each cluster c_i, each entity pair $p_{ij} \in c_i$ holds the same semantic relation. To label the extracted relations, the Relation Tagging module employs a closed frequent sequence mining algorithm to identify the closed frequent sequential patterns in all occurring sentences where the entity pairs of each cluster appear. Then we use these extracted patterns to label and describe the semantic relation held in each cluster. Finally, in order to extract the inter-sentential relations, the Anaphora Resolution module applies the pronominal anaphora resolution algorithm to the documents and obtains the substitution text where pronominal references are substituted by the noun phrases they refer to. Subsequently, REACTOR could use the other four modules introduced above to process the substitution text to extract the single-sentential and inter-sentential relations together.

3.2. Seed Extractor

It is difficult for non-expert users to provide human-selected seeds or manually tagged training examples that are expensive and need significant human effort. The proposed REACTOR adopts a statistical method to extract some representative entity pairs as seeds to train the classifier automatically.

Figure 2. The REACTOR architecture



Obviously, to train the classifier, the extracted seeds should contain both positive examples, in which the entity pairs are almost likely to be related and the two entities of each entity pair depend on each other heavily, and negative examples in which the entity pairs are unrelated and the two entities of each entity pair are independent of each other. Therefore, we need to define a weighting function that can assign a weight to each entity pair and indicate how strongly the two entities of the entity pair are related. Intuitively, the entity that is strongly related to entity e_i should be the one that frequently co-occurs with entity e_i, but infrequently co-occurs with others.

Many weighting functions can be used to measure the dependency between two entities. Previous research on statistical natural language processing has proved that co-occurrence statistics are highly informative and simple when computed over large corpora (Banko & Brill, 2001). Despite that, co-occurrences may not be a good measure for our task whose problem is that co-occurrence statistics have a strong bias towards global common entity pairs in the collection. For example, entity e_j co-occurs with entity e_i many times, but the entity pair $< e_i, e_j$

> may still not be a good positive seed if the entity e_j also co-occurs with other entities frequently. Therefore, we should penalize these entities by dividing by their occurrence frequencies, which is like the computation of the *pointwise mutual information* (PMI). The PMI value of an entity pair is computed as:

$$I\left(e_i, e_j\right) = log_2 \frac{P\left(e_i,\ e_j\right)}{P\left(e_i\right)P\left(e_j\right)} \tag{1}$$

where $P\left(e_i,\ e_j\right)$ is the co-occurrence probability of entities e_i and e_j, $P\left(e_i\right)$ and $P\left(e_j\right)$ are the occurrence probabilities of entity e_i and entity e_j respectively in the corpus.

However, the PMI value has a strong bias towards low frequent entity pairs. For example, entities e_i and e_j just appear in the corpus once respectively and moreover, they happen to co-occur within the same sentence. In this situation, the PMI value of this entity pair <e_i, e_j> is extremely large so that we consider it as a positive seed. But entities e_i and e_j are likely to be unrelated since they just co-occur by chance.

Therefore, in order to avoid the bias mentioned above, we compute the relatedness weight for each entity pair e_i, e_j as follows:

$$weight\left(e_i, e_j\right) = C\left(e_i,\ e_j\right) log_2 \frac{P\left(e_i,\ e_j\right)}{P\left(e_i\right)P\left(e_j\right)} \tag{2}$$

where $C\left(e_i,\ e_j\right)$ is the number of co-occurrences of entities e_i and e_j, $P\left(e_i,\ e_j\right)$ is the co-occurrence probability of entities e_i and e_j, $P\left(e_i\right)$ and $P\left(e_j\right)$ are the occurrence probabilities of entity e_i and entity e_j respectively in the corpus.

For example, we want to compute the relatedness weight for entities “Jane” and “HP Labs China” in PEO-ORG domain. The number of occurrences of entity “Jane” in the corpus is 13, while the number of occurrences of entity “HP Labs China” is 298. And the number of co-occurrences of these two entities is 8. The total number of occurrences of entity pairs in PEO-ORG domain is 12038, while the total numbers of occurrences of entities with types PEO and ORG are 6010021 and 3819621, respectively. The relatedness weight for this entity pair “Jane” and “HP Labs China” can be computed as

$$8*log_2 \frac{8/12038}{13*298/\left(6010021*3819621\right)} = 175.27\ .$$

For the entity pair “Owen” and “HP Labs China”, the number of occurrences of entity “Owen” in the corpus is 24 and the number of co-occurrences of these two entities is 1. The relatedness weight for this entity pair “Owen” and “HP Labs China” can be computed as

$$1*log_2 \frac{1/12038}{24*298/(6010021*3819621)} = 18.02 .$$

The relatedness weight for the entity pair "Jane" and "HP Labs China" is much larger than it for the entity pair "Owen" and "HP Labs China". Therefore, the entity pair "Jane" and "HP Labs China" is more likely to be related compared with the entity pair "Owen" and "HP Labs China".

Let p_{ij} beanentitypair$<e_i, e_j>$thatoccurswithinonesentenceand $P = \{p_{11}, p_{12}, \ldots, p_{ij}, \ldots\}$ be the set of all such entity pairs in the corpus. According to Equation 2, we can calculate the relatedness weight w_{ij} for each entity pair p_{ij} which can tell us how strongly the entity pair p_{ij} is related. Then, the task of extracting positive seeds is to extract a subset of *k* entity pairs $P' \subseteq P$, such that $\forall\, p_{ij} \in P'$ and $\forall\, p_{sv} \in P - P'$, we have $w_{ij} \geq w_{sv}$. On the contrary, the task of extracting negative seeds is to extract a subset of *s* entity pairs $P'' \subseteq P$, such that $\forall\, p_{ij} \in P''$ and $\forall\, p_{sv} \in P - P''$, we have $w_{ij} \leq w_{sv}$.

With the relatedness weighting function defined in Equation 2, we can compute the relatedness weight $w_{ij} = weight(p_{ij})$ for each entity pair p_{ij}. Then rank $p_{ij} \in P$ with respect to w_{ij} in descending order and select the top *k* p_{ij}'s as the positive seeds, and in turn rank $p_{ij} \in P$ with respect to w_{ij} in ascending order and select the top *s* p_{ij}'s as the negative seeds. The two parameters *k* and *s* are specified empirically.

3.3. Relation Detection

The enterprise data has much lower redundancy in comparison with the Web data, and most of entity pairs only occur a few times in the entire data set shown in Figure 1, which makes the bootstrapping methods (Agichtein & Gravano, 2000 ; Zhu et al., 2009 ; Brin, 1998 ; Etzioni et al., 2005) hard to be applied. Therefore, we leverage a classifier to detect the related entity pairs with any occurrence frequency according to the contexts where the entity pairs appear, rather than using a simple frequency threshold to filter the low frequent entity pairs that is used by some relation extraction systems (Banko et al., 2007 ; Hasegawa et al., 2004 ; Gonzàlez & Turmo, 2009 ; Bollegala, Matsuo, & Ishizuka, 2010).

Starting from the seed set $S = \{s_1, s_2, \ldots, s_{(k+s)}\}$, where $s_i \in S$ is a positive or negative training seed provided by the Seed Extractor, the goal of this stage is to train the classifier and label each entity pair $<e_i, e_j>$ with the targeting types T_m and T_n occurring within the same sentence as related or not.

Each occurrence of the seed $s_i \in S$ is considered as a training tuple. Here, we represent each co-occurring sentence $st_i =< t_{i1,}\ t_{i2,}\ t_{i3}, \ldots >$ as a sequence of tokens t_{iq} in the sentence. We also define some domain-independent features that can be used to capture the syntactic information and the entity information for each sentence where the entity pair occurs. We can map each occurrence of the entity pair to a feature vector $< x_{i1}, x_{i2}, \ldots >$ where x_{ir} tells the value of the *r*th feature of the co-occurring sentence st_i. The feature vector can be changed by adding or eliminating some features easily, which gives flexibility to the model in an efficient and simple way. The features used in our experiment are listed as follows:

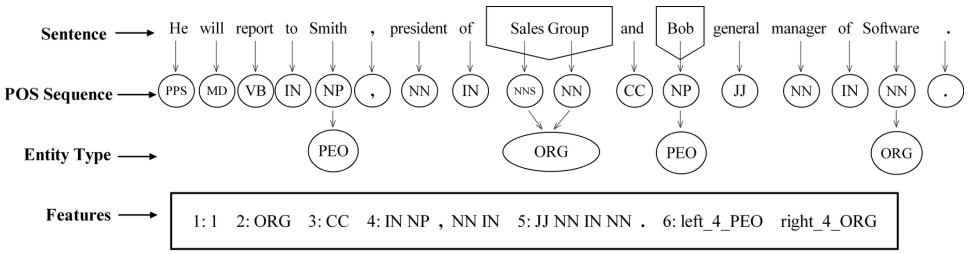
1. The number of tokens between the entities in the sentence;
2. The type of the entity that appears first in the sentence;
3. The part-of-speech tag sequence between the entities;
4. The part-of-speech tag sequence before the first entity within distance *d*;
5. The part-of-speech tag sequence after the second entity within distance *d*;
6. The position and type of the other entities in the sentence.

An example of the feature generation for a certain entity pair with T_m = PEO and T_n = ORG (Here, PEO means PEOPLE and ORG means ORGANIZATION), is shown in Figure 3. In the example, the two considered entities are "Bob" with entity type of PEO and "Sales Group" with entity type of ORG. Firstly, the value of Feature 1 is 1 because the number of tokens between "Bob" and "Sales Group" is 1 and the value of Feature 2 is ORG since the entity "Sales Group" with type of ORG appears first in the sentence compared with the entity "Bob". The Features 3, 4, and 5 are the part-of-speech tag sequences respectively between entities, before "Sales Group" within distance *d* and after "Bob" within distance *d*. In our experiments, the performance of the classifier is insensitive to the parameter *d* and we obtain very similar classification results when d is varied from 3 to $+\infty$. In order to reduce the number of features for the SVM classifier for the purpose of efficiency, we set *d* to 5 in the experiments. Finally, as there is an entity "Smith" with type of PEO before the former considered entity "Sales Group" at the fourth position and an entity "Software" with type of ORG after the latter considered entity "Bob" at the fourth position, Feature 6 activates features left_4_PEO and right_4_ORG respectively. As we can see, the entities "Bob" and "Sales Group" in this example are unrelated, and those generated features including the syntactic and entity information of the context can support the classifier to label them correctly.

To extract the related entity pairs, firstly, we use the feature vectors produced from the seed set *S* to train the classifier which is *libsvm*[2] used in the experiment. We use the tool in *libsvm* which does the whole classification procedure including scaling and model selection completely automatically. Then we use OpenNLP toolkits to annotate each sentence in the entire corpus with POS tags. Finally, each occurrence of the entity pair is presented to the trained classifier and the classifier labels each of them as related or unrelated. Since each entity pair may have more than one occurrence, we consider the entity pair as related if and only if the number of occurrences classified as related for this entity pair is larger than or equal to the number of occurrences classified as unrelated.

In the process of relation detection, we only use the shallow linguistic processing technique (i.e., part-of-speech tagging). In contrast to deep natural language processing techniques used by

Figure 3. Example of feature generation (in PEO-ORG domain)



many extraction systems (Agichtein & Gravano, 2000 ; Kambhatla, 2004 ; Etzioni et al., 2005 ; Banko et al., 2007 ; Shinyama & Sekine, 2006), shallow NLP techniques are more robust and efficient, which is very important for the relation extraction over the large-scale enterprise data.

It is also noteworthy that differently from some relation extraction systems (Banko et al., 2007 ; Hasegawa et al., 2004 ; Gonzàlez & Turmo, 2009 ; Bollegala et al., 2010) that have a frequency threshold to filter the low frequent entity pairs or a distance threshold to filter the entity pairs with a long distance in the sentence, REACTOR can extract the related entity pair with any frequency and any distance within one sentence, which can significantly improve the performance of relation extraction on the low-redundancy enterprise data.

3.4. Relation Categorization

After the classification, we obtain all related entity pairs. As we do not know any prior knowledge about the number and types of the relations existing in the corpus, therefore, it is extremely useful to identify the semantic relations between entities. To extract the semantic relations, we assume that entity pairs occurring in the similar context likely have the same semantic relation and can be clustered into a group. Entity pairs in each group produced by the clustering algorithm are expected to express the same semantic relation.

We first adopt a vector space model (Salton, Wong, & Yang, 1975) to represent the context of an entity pair. For each entity pair, we firstly obtain all their occurrence sentences and eliminate some non-essential phrases, such as stop words, prepositional phrases and modifiers, from these sentences. Meanwhile, we filter out the other entities appearing in the sentences as well, because these words do not express any semantic relation and would introduce much noise in calculating similarities. In constructing the context vector, we consider not only the bag of words between the entities but also those around the entities in each occurrence sentence within the same distance *d* as introduced in Section 3.3. These words are stemmed by Porter Stemmer[3] and are weighted in the context vector by their term frequency empirically (Different term weighting strategies will be discussed in Section 4.2).

For example, for the entity pair "Jane" and "HP Labs China", one of their occurrence sentences is "Michael reports to Jane, who is the project manager of HP Labs China". Before generating the context vector for this occurrence, we eliminate the stop words such as "to", "who", "is", "the" and "of". We also filter out the entity "Michael" because this entity mention does not express any semantic relation between entities "Jane" and "HP Labs China". Therefore, we obtain the context vector {"reports", "project", "manager"} before stemming and weighting.

After generating the context vector for each entity pair, we introduce a similarity function to measure the similarity between any two context vectors and then adopt a clustering algorithm to further group the entity pairs. Finally, the entity pairs clustered into the same group are expected to have the same semantic relation.

Cosine is widely used to compute the similarity between two vectors and is well applied in the information retrieval field. In our approach, we use cosine value of two context vectors to measure the semantic similarity of two corresponding entity pairs. Generally, the cosine similarity $sim\left(c\left(\acute{a}\right),\ c\left(\hat{a}\right)\right)$ of two context vectors $c\left(\acute{a}\right)$ and $c\left(\hat{a}\right)$ is computed as:

$$sim\left(c\left(\acute{a}\right),\ c\left(\hat{a}\right)\right) = \frac{\sum_{i=1}^{k} a_i * b_i}{\sqrt{\sum_{i=1}^{k} a_i^2} * \sqrt{\sum_{i=1}^{k} b_i^2}} \tag{3}$$

where $c(\text{á}) =< a_1, a_2, \ldots, a_k >$ and $c(\text{â}) =< b_1, b_2, \ldots, b_k >$.

With cosine similarity, we expect to group the entity pairs such that the similarity within intra-cluster is high and that between inter-clusters is low. As the number of relations is unknown beforehand, we adopt the hierarchical clustering algorithm. This clustering algorithm does not require to pre-define the number of clusters and the result of the clustering is independent of the order of entity pairs. The algorithm iteratively groups two clusters of entity pairs with the maximum similarity, where the similarity between two clusters is defined as the cosine similarity between the furthest entity pairs in the two clusters empirically (Different cluster distance computation strategies will be discussed in Section 4.2). The algorithm terminates when the maximum similarity between any two clusters becomes smaller than a pre-defined threshold $\tilde{\text{a}}$. The details of the algorithm are shown in Figure 4.

3.5. Relation Tagging

Although the entity pairs are clustered into a set of groups each of which represents a type of semantic relation between entities, we do not know the exact semantic relation held in each cluster. For the evaluation and presentation purpose, it is extremely useful and important to label clusters with some representative tags to describe the semantic relations existing in them.

We represent each co-occurring sentence $st_i =< t_{i1,}\ t_{i2,}\ t_{i3}, \ldots >$ as a sequence of tokens t_{ij} in the sentence. For each co-occurring sentence st_i where the entity pair with types T_m and T_n appears, we replace the two corresponding entities with two variables T_m and T_n respectively to produce st_i'. The tokens which belong to the entity with the type of T_m are replaced by T_m, whereas the tokens which belong to the entity with the type of T_n are replaced by T_n. Then we construct the sequence database $D = \left\{st_1', st_2', \ldots, st_i', \ldots\right\}$ for each cluster.

Let pt_i be a subsequence in *D* and we denote the set of sequences in which pt_i appears as $D_i = \{st_\alpha' | pt_i \in st_\alpha', st_\alpha' \in D\}$. Now, we give out the definition of frequent sequence pattern.

Definition 1 (Frequent Sequence Pattern): A sequence pattern pt_i is *frequent* in a database *D*, if $\frac{|D_i|}{|D|} \geq \sigma$, where σ is a pre-defined threshold and $\frac{|D_i|}{|D|}$ is called *relative support* of pt_i.

We know that the frequent sequence patterns are the subsequences that appear in the data set frequently and moreover, all subsequences of a long frequent sequence pattern must be frequent due to the downward closure property (Agrawal & Srikant, 1994). Thus, the set of frequent sequence patterns has redundancy of sequences caused by the inclusion of both a frequent sequence pattern and its subsequences. Therefore, in our work, we use closed frequent sequence patterns to label the semantic relation. The definition of closed frequent sequence pattern is shown as follows.

Definition 2 (Closed Frequent Sequence Pattern): A frequent sequence pattern pt_i is closed if and only if there exists no super-sequence pt_j of pt_i, s.t. $D_i = D_j$.

Figure 4. The hierarchical clustering algorithm

Input: A set of n entity pairs: $P = \{p_1, p_2, \ldots, p_n\}$, Threshold of similarity: γ.
Output: A set of clusters: $C = \{C_1, C_2, \ldots, C_k\}$.

1: Initialize n clusters C_i, each as an entity pair p_i
2: Compute the cosine similarity s_{ij} between entity pairs in P
3: Set the current maximum similarity $s = \max(s_{ij})$
4: **while** $(s > \gamma)$ **do**
5: Select s_{lt} where $(l, t) = \arg\max_{i,j} s_{ij}$
6: Merge clusters C_l and C_t into a new cluster C_u
7: $s \leftarrow s_{lt}$
8: **for all** $C_v \neq C_u$ **do**
9: Compute $s_{uv} = \min(s_{\alpha\beta})$ where $p_\alpha \in C_u$, $p_\beta \in C_v$
10: **end for**
11: **end while**

To extract the closed frequent sequence pattern, we employ the BIDE algorithm (Wang, Han, & Li, 2007) that can efficiently discover closed frequent sequence patterns without candidate maintenance and test. To express the semantic relation between entities with types T_m and T_n respectively, we only retain the closed frequent sequence patterns that contain both T_m and T_n in their sequences.

Unlike the system proposed in Hasegawa et al. (2004) that just simply selects the most frequent common words to label the extracted relation clusters, we use the closed frequent sequence patterns which can retain the inherent syntactic structure of the sentences where the semantic relations are mentioned and can describe the semantic relation more accurately, which can be seen from the experimental results shown in Section 4.2.

3.6. Anaphora Resolution

The analysis in Swampillai and Stevenson (2010) has shown that some inter-sentential relations are commonly asserted using anaphoric expressions. Therefore, it seems reasonable to extract

inter-sentential relations by solving pronominal references. We use JavaRAP[4] which is a Java-based implementation of the seminal Resolution of Anaphora Procedure (RAP) algorithm (Lappin & Leass, 1994) to resolve the pronominal anaphora. JavaRAP can identify both inter-sentential and intra-sentential antecedents of third person pronouns and lexical anaphors (Qiu, Kan, & Chua, 2004). It takes the parsed sentences as input, and generates a list of anaphora-antecedent pairs as output. Alternately, it can produce an in-place substitution of the anaphors with their antecedents.

For example, given the input sentence "Neal recently had a talk in Austin, TX. He is Senior Vice President of HP Software.", JavaRAP can produce an in-place substitution of the anaphor (i.e., "He") with its antecedent (i.e., "Neal") and output the substitution sentence "Neal recently had a talk in Austin, TX. Neal is Senior Vice President of HP Software.". Hence, we transform the inter-sentential relation between entities "Neal" and "HP Software" in PEO-ORG domain asserted using anaphoric expression into the single-sentential relation via solving the pronominal reference.

In order to extract the inter-sentential relations, we use a module called Anaphora Resolution to apply JavaRAP to the parsed sentences of documents produced by the OpenNLP toolkits and obtain the substitution text where pronominal references are substituted by the noun phrases they refer to. Subsequently, REACTOR leverages the other four modules introduced above to process the substitution text to extract the single-sentential and inter-sentential relations together.

4. EXPERIMENTS

To evaluate the effectiveness of REACTOR, we tested it on a real-world enterprise data set from HP Company whose details are given in Section 4.1. We compared REACTOR with a clustering-based method proposed in Hasegawa et al. (2004). We chose this method as the baseline method because only this existing method can extract different semantic relations appearing in one type of entity pair, which is quite similar to REACTOR. Benefits obtained by applying pronominal anaphora resolution are measured by comparing the system performance with and without taking into account information referenced pronominally. In Section 4.2, we present the experimental results, which show that REACTOR achieves significantly higher precision and recall over the enterprise data and can label each extracted semantic relation with tags more accurately compared with the baseline method, meanwhile, pronominal anaphora resolution can improve the system performance greatly.

4.1. Data Sets

Our experiments were conducted on a large real-world enterprise data set from HP Company in which there are over three million pages including e-mails, internal Web pages, and word processing files. In the data set, about 67% of documents are internal Web pages, about 28% of documents are emails, while the other 5% of documents are word processing files. The average size of these documents is 16.7k and these documents are selected from May 1, 2008 to September 30, 2008 as a subset of all documents within HP. In this data set, there are about 97051 different entities with the type of people, 916 distinct organization entities and 2123 distinct product entities. Moreover, these types of entities have been discovered in advance as the input of our framework.

4.2. Methods and Results

In this subsection, we present the evaluation results of our REACTOR. The usual metrics of Precision (P), Recall (R) and *F*-score (F) on the classification and clustering results are used to evaluate the performance of REACTOR. For evaluation purpose, we determined the relation that exists among most entity pairs in one cluster as the major relation of this cluster. The entity pairs having the major relation of this cluster are considered as correct pairs, otherwise, they are considered as incorrect pairs. Furthermore, we only considered the clusters consisting of two or more pairs in the same way as that in Hasegawa et al. (2004). In the experiments, we considered the relations in two different domains. One is the PEOPLE-ORGANIZATION (PEO-ORG) domain and another is the PEOPLE-PRODUCT (PEO-PRO) domain. We set k to 100 and s to 70 for PEO-ORG domain and k to 100 and s to 50 for PEO-PRO domain empirically when we selected the seeds in the Seed Extractor module (k represents the number of positive seeds and s represents the number of negative seeds). Meanwhile, the performance of REACTOR is not very sensitive to these two parameters, because these two parameters just affect the training of the classifier. When k is set from 60 to 250, and s is set from 40 to 150 in PEO-ORG domain, and k is set from 60 to 200, and s is set from 30 to 90 in PEO-PRO domain, the classification results are very similar in the experiments we conducted.

Firstly, we evaluated the classification results of the Relation Detection module for both domains. We randomly selected 500 entity pair occurrences in PEO-ORG domain and 250 entity pair occurrences in PEO-PRO domain. The selected occurrences of entity pairs are manually labeled as related or not according to their contexts. There are total 371 occurrences of entity pairs labeled as related in PEO-ORG domain and 203 occurrences of entity pairs labeled as related in PEO-PRO domain. The experimental results of classification are shown in Table 1. From the results we can see that our Relation Detection module can achieve significantly high *F*-score for both domains, which is beneficial for the next module Relation Categorization.

Then, in order to evaluate the clustering results of the Relation Categorization module, we created a test data set for clustering and manually labeled semantic relations in this test data set. Since for the large data set, labeling all relations is impractical and it is difficult to quantitatively evaluate REACTOR over the entire data set. To compare REACTOR with the baseline method and give a quantitative analysis, we randomly selected 500 and 250 entity pairs (not entity pair occurrences) classified as related by our Relation Detection module in PEO-ORG domain and PEO-PRO domain respectively as the test data set for clustering, as our Relation Categorization module clusters entity pairs nor entity pair occurrences. We analyzed the test data set and manually labeled the entity pairs into 53 different semantic relations in PEO-ORG domain and 47 different semantic relations in PEO-PRO domain both including a type of "No Relation" which means the two corresponding entities are unrelated. The types of relations and the number of entity pairs in each semantic relation for PEO-ORG domain and PEO-PRO domain are shown in Table 2 and Table 3, respectively. Due to limited space, we do not list all semantic relations and we put the number of entity pairs of the semantic relations that do not appear in Table 2 and

Table 1. Experimental results of classification for both domains

Domain	P	R	F
PEO-ORG	0.873	0.960	0.914
PEO-PRO	0.856	0.943	0.897

Table 3 into the relation "Others" in the tables for both domains. The relation "Employee of" for PEO-ORG domain in Table 2 means that for each entity pair of the relation "Employee of" there is no context in all its occurrences indicating the concrete semantic relation between the entities, but we can judge these two entities are related according their context. For example, for the sentence "Bob of Sales Group talked to …", we can predict that "Bob" is an employee of the "Sales Group", but we do not know the concrete role of "Bob" in this organization. Therefore, we defined this type of entity relationship as "Employee of". Meanwhile, the relation "Related" for PEO-PRO domain in Table 3 has the similar meaning. From the distribution of semantic relations in Table 2, we can see that almost half of the entity pairs in PEO-ORG domain are annotated as "Employee of" that means there is no concrete semantic relation indicated by the context. On the other hand, the number of entity pairs annotated as "Related" for PEO-PRO domain in Table 3 is much less. Since our proposed method aims to extract semantic relations between entities, when we evaluated our approach and the baseline method, we did not consider the entity pairs which are annotated as "Employee of" for PEO-ORG domain in Table 2 and "Related" for PEO-PRO domain in Table 3.

We evaluated the performance of REACTOR with different term weighting and linkage strategies for clustering in the Relation Categorization module. Table 4 presents the experimental results of REACTOR under different strategy combinations in PEO-ORG domain with the optimal clustering threshold. As hierarchical clustering algorithm needs a pre-defined threshold to terminate the clustering process, the clustering result varies with different pre-defined thresholds. The optimal clustering threshold is the pre-defined threshold generating the best clustering result. Three different cluster distance computation methods (i.e., single linkage, average linkage, and complete linkage) and two different term weighting strategies for generating context vectors (i.e., tf and tf*idf) are compared using all possible combinations. Here tf means the term frequency and idf means the inverse document frequency, both of which are widely used in information retrieval. The results in Table 4 show that when we use complete linkage for clustering and tf as term weighting for generating context vectors, we can achieve the best result compared with the other strategy combinations. The tf term weighting is better than tf*idf in relation clustering since the terms retained to generate the context vector are close to the entities in the sentences and most of them indicate the semantic meaning of the relations, meanwhile, those indicating

Table 2. Manually labeled semantic relations in test data set for PEO-ORG domain

Semantic relation	Leader	Vice President	Director	General Manager
# of pairs	45	26	25	24
Semantic relation	Manager	Technology Consultant	Solution Architect	Program Manager
# of pairs	17	10	9	7
Semantic relation	Project Manager	CTO	Operations Manager	Engineer
# of pairs	7	6	5	5
Semantic relation	Researcher	Product Manager	Developer	Marketing Manager
# of pairs	4	3	2	2
Semantic relation	Presales Manager	Client Manager	Account Manager	Assistant
# of pairs	2	2	2	2
Semantic relation	IT Manager	Employee of	Others	No Relation
# of pairs	2	242	24	27

Table 3. Manually labeled semantic relations in test data set for PEO-PRO domain

Semantic relation	Change Proposal	Technical Specialist	Manager	Product Manager
# of pairs	48	3	9	8
Semantic relation	Solutions Architect	Technical Consultant	Support	Achieve Certification
# of pairs	7	7	6	6
Semantic relation	Program Manager	Project Manager	Demonstrate	Storage Consultant
# of pairs	4	4	3	2
Semantic relation	Test	Use	Win	Related
# of pairs	11	6	24	25
Semantic relation	Others	No Relation		
# of pairs	55	22		

Table 4. Performance of REACTOR with different term weighting and linkage strategies for clustering in PEO-ORG domain

	Threshold	P	R	F
Single+tf*idf	0.0080	0.475	0.569	0.518
Average+tf*idf	0.0016	0.581	0.592	0.586
Complete+tf*idf	0.0007	0.733	0.659	0.694
Single+tf	0.56	0.532	0.613	0.569
Average+tf	0.34	0.771	0.711	0.740
Complete+tf	0.24	**0.795**	**0.819**	**0.807**

words are relatively frequent in the context vectors compared with other special words existing in each vector. Thus, if we use tf*idf as term weighting, the relative weight of other special words in each context vector will increase and the relative weight of indicating words will decrease because the document frequency of the special words is much smaller than the indicating words. In this case the entity pairs are possible to be merged into one cluster due to the special words consequently, which leads to the dissatisfactory clustering results.

We also compared REACTOR with the baseline method under different configurations in the two different domains for clustering over the test data set. The baseline method needs three parameters including maximum context word length, the occurrence frequency threshold of entity pairs, and the norm threshold for context vectors to filter the unreliable pairs. If we use the original setting of these thresholds introduced in Hasegawa et al. (2004), all entity pairs in the test data set will be filtered out and no entity pair is retained to start the clustering process, which also strongly reveals the low redundancy of the enterprise data and that the method based on the high redundancy of the Web corpus is not suitable to be applied to the enterprise data set. Thereby, to compare REACTOR with the baseline method, we must change the threshold setting of the baseline method. The simplest way is to directly eliminate those thresholds and all entity pairs are retained for the clustering process which we refer to Baseline. We also selected the optimal thresholds for the baseline method which can obtain the best *F*-score. This method is referred to Ba-Optimal. The optimal thresholds are 10 in PEO-ORG domain and 15 in PEO-

Table 5. Experimental results of REACTOR and the baseline methods for clustering in both domains

Domain	Method	Threshold	P	R	F
PEO-ORG	Baseline	0.0001	0.517	0.601	0.556
	Ba-Optimal	0.0003	0.638	0.549	0.590
	REACTOR	0.24	**0.795**	**0.819**	**0.807**
PEO-PRO	Baseline	0.0005	0.608	0.718	0.659
	Ba-Optimal	0.0010	0.696	0.670	0.683
	REACTOR	0.26	**0.846**	**0.729**	**0.783**

PRO domain for the maximum context word length, and 0 for both the occurrence frequency threshold and the norm threshold for both domains. Table 5 shows the experimental results of REACTOR and the two baseline methods with the optimal clustering thresholds in two different domains. Our proposed approach REACTOR uses complete linkage for clustering and tf for term weighting. It can be seen from the results that the overall Precision, Recall and *F*-score of REACTOR are significantly better than both Baseline and Ba-Optimal in two different relation extraction tasks. Despite that different clustering thresholds can generate different results, we find that if we choose certain linkage strategy and term weighting (e.g., complete linkage+tf), the performance of REACTOR is not very sensitive to the threshold. For PEO-ORG domain, when the pre-defined clustering threshold is set from 0.20 to 0.33, the *F*-score of REACTOR is varied from 0.786 to 0.807, and when the threshold is set to 0.24, REACTOR obtains the best *F*-score (i.e., 0.807). Meanwhile, for PEO-PRO domain, when the pre-defined clustering threshold is set from 0.20 to 0.33, the *F*-score of REACTOR is varied from 0.760 to 0.783, and when the threshold is set to 0.26, REACTOR obtains the best *F*-score (i.e., 0.783).

Then, we investigated the performance of our Relation Tagging module, which labels each cluster with tags to describe the semantic relation. To select the representative tags, our model REACTOR firstly replaces the two considered entities in all co-occurring sentences with "PEO" or "ORG" in PEO-ORG domain and "PEO" or "PRO" in PEO-PRO domain respectively according to their entity types. Next, REACTOR runs the BIDE algorithm on the sequence database consisting of all co-occurring sentences for each cluster and sets the relative support threshold to 0.5. Then, REACTOR only retains the closed frequent sequences that contain both "PEO" and "ORG" in PEO-ORG domain or "PEO" and "PRO" in PEO-PRO domain. Table 6 shows a part of clusters for each domain, along with their ratio of the major relation in each cluster following the name of the relation. We also show the tagging results of REACTOR for each cluster and their relative support within the bracket following each selected tag. To compare REACTOR with the baseline method which simply selects the most frequent words between entities in the sentences to label the relation, we also list the labeling results of the baseline method as well as their relative frequency within the bracket following each selected tag in Table 6. The labeling results in PEO-ORG domain are presented on the top of the table and results in PEO-PRO domain are on the bottom of the table. From the tagging results we can see that REACTOR can label the clusters more accurately than the baseline method. Specially, when the semantic relation is not mentioned between entities in the sentence, but around the entities such as the relation "Change Proposal" and "Test" in PEO-PRO domain, the baseline method cannot extract the accurate words that express the semantic relation. Furthermore, our framework can retain the inherent syntactic structure of the sentences where the semantic relation is mentioned and describe the targeting semantic relation more concretely than the baseline method.

Table 6. A part of generated clusters and their relation tagging results for each domain

Major relations	Ratio	REACTOR-Tags(Relative support)	Baseline-Tags(Relative frequency)
Vice President	20/20	PEO vice president ORG (0.772)	vice (0.794), president (0.794) marketing (0.513), senior (0.136)
		PEO vice president of ORG (0.557)	
		PEO vice president for ORG (0.5)	
General Manager	14/14	PEO general manager ORG (0.734)	president (0.776),vice (0.776)
		PEO vice president ORG (0.71)	general (0.735), manager (0.735)
		PEO vice president general manager ORG (0.69)	senior (0.327), workstations (0.204)
Operation Manager	5/6	PEO manager ORG (0.846)	operation (0.846), manager (0.846)
		PEO operation manager ORG (0.538)	solution (0.462), trading (0.462)
Leader	9/9	ORG led by PEO (0.722)	led (0.778), by (0.722), design (0.167)
Director	9/11	PEO director ORG (0.696)	director (0.878), business (0.683)
Technical Consultant	6/6	PEO technical consultant ORG (0.976)	consultant (0.977), technical (0.977)
		PEO senior technical consultant ORG (0.512)	senior (0.512), workstation (0.349)
Program Manager	4/4	PEO program manager ORG (1.0)	manager (1.0), program (1.0)
Project Manager	3/3	PEO project manager ORG (1.0)	project (1.0), manager (1.0)
Researcher	2/3	PEO research analyst ORG (0.576)	research (0.615), analyst (0.576)
CTO	2/2	ORG CTO PEO (0.6)	CTO (0.6), innovation (0.2)
Change Proposal	45/45	proposal was changed by PEO product PRO (0.903)	product (0.923), using (0.395)
		source proposal was changed by PEO PRO (0.875)	source (0.167), proposal (0.0625)
Solutions Architect	5/5	PEO architect PRO (1.0)	architect (1.0), storage (0.928)
		PEO storage architect PRO (0.929)	solutions (0.821), solution (0.178)
		PEO solutions architect PRO (0.821)	senior (0.035), engineer (0.035)
Technical Consultant	4/4	PRO PEO technical consultant (0.933)	regards (0.933), consultant (0.066)
Test	4/4	have already tested the PRO PEO (1.0)	met (1.0)
		have already tested the PRO PEO at the (0.5)	tomas (0.5), martin (0.5)
Use	3/3	employee PEO using an PRO (1.0)	using (1.0), employee (0.333), fort (0.333)
		taken by employee PEO using an PRO (0.667)	
Storage Consultant	2/2	PEO storage PRO (1.0)	storage (1.0), consultant (0.789)
		PEO storage consultant PRO (0.789)	

To measure the benefits obtained by solving pronominal references, we compared system performance with and without pronominal anaphora resolution. Due to the impracticalness to give quantitative evaluation over the entire data set, we sampled twenty thousand pages as the test corpus and evaluated the system performance of semantic relation extraction over this corpus in PEO-ORG domain. Currently JavaRAP only considers noun phrases contained within three sentences proceeding the anaphor and those in the sentence where the anaphor resides (Qiu et al., 2004). However, from the analysis in Swampillai and Stevenson (2010) we can see about 76.3% of inter-sentential relations are contained within a window of four sentences. Hence, the limitation of JavaRAP has little impact on the system performance. Table 7 shows the performance of REACTOR with and without pronominal anaphora resolution in PEO-ORG domain over the test corpus. From Table 7 we see that with the same optimal clustering threshold, REACTOR with and without pronominal anaphora resolution get almost the same precision. Despite that, the number of correct entity pairs REACTOR extracts from the two thousand pages increases by 21.2% (i.e., from 435 to 527) after REACTOR makes use of the information referenced pronominally. As the results show, we can say that solving pronominal references improves REACTOR's performance with high level of recall.

5. CONCLUSION

The existing relation extraction methods, which work on the Web data very well, are not suitable for relation extraction on the low-redundancy enterprise data. In this paper, we propose a novel hybrid semantic relation extraction framework called REACTOR, which combines a statistical method, classification, and clustering techniques to extract relations from the enterprise data. A statistical method is introduced to extract a set of representative entity pairs containing both positive and negative examples for the classifier. Then REACTOR employs a classifier to extract all related entity pairs by defining some domain-independent features. A clustering algorithm is used to identify the semantic relation between each pair of entities. To label the extracted semantic relation, REACTOR exploits a closed frequent sequence mining algorithm to extract the representative tags to describe the relationship in each cluster. Meanwhile, REACTOR applies pronominal anaphora resolution to extract more relations expressed across sentence boundaries. Finally, we evaluate REACTOR on a large real-world enterprise corpus from HP and the results show that REACTOR can extract the semantic relation more effectively in comparison with the baseline method, and the extracted tags can describe the semantic relation more accurately. Moreover, as the results show, the application of anaphora resolution improves REACTOR's performance greatly and seems to be very essential for inter-sentential relation extraction.

Table 7. System performance with and without pronominal anaphora resolution in PEO-ORG domain

	Threshold	Correct Pairs	Total Pairs	Precision
REACTOR without anaphora resolution	0.21	435	536	0.812
REACTOR with anaphora resolution	0.21	527	650	0.811

ACKNOWLEDGMENT

This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2014CB340505, National Natural Science Foundation of China under Grant No. 61532010, 61272088 and 61502253, and Tsinghua University Initiative Scientific Research Program. Ping Luo was supported by the National Natural Science Foundation of China (No. 61473274), National High-tech R&D Program of China (863 Program) (No. 2014AA015105). The corresponding author is Jianyong Wang.

REFERENCES

Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. *Proceedings of the fifth ACM Conference on Digital Libraries (DL'00)* (pp. 85-94). doi:10.1145/336597.336644

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)* (pp. 487-499).

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL'01)* (pp. 26-33). doi:10.3115/1073012.1073017

Banko, M., Cafarella, M. J., Soderl, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)* (pp. 2670-2676).

Banko, M., & Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics (ACL'08)* (pp. 28-36).

Bollegala, D. T., Matsuo, Y., & Ishizuka, M. (2010). Relational duality: Unsupervised extraction of semantic relations between entities on the web. *Proceedings of the 19th International Conference on World Wide Web (WWW'10)* (pp. 151-160). doi:10.1145/1772690.1772707

Brin, S. (1998). Extracting patterns and relations from the world wide web. *Proceedings of the International Workshop on the World Wide Web and Databases (WebDB'98)* (pp. 172-183).

Crowley, C., & Harris, S. (2007). The Sedona conference glossary: E-discovery and digital information management (2nd edition). *Proceedings of the Sedona Conference 2007*.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., & Yates, A. et al. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, *165*(1), 91–134. doi:10.1016/j.artint.2005.03.001

Freitag, D., & Mccallum, A. K. (1999). Information extraction with hmms and shrinkage. *Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction* (pp. 31–36).

Giuliano, C., Lavelli, A., & Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)* (pp. 401-408).

Gonzàlez, E., & Turmo, J. (2009). Unsupervised relation extraction by massive clustering. *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining (ICDM '09)* (pp. 782-787). doi:10.1109/ICDM.2009.81

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. *Proceedings of 16th Conference on Computational Linguistics (COLING'96)* (pp. 466-471). doi:10.3115/992628.992709

Hasegawa, T., Sekine, S., & Grishman, R. (2004). Discovering relations among named entities from large corpora. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL'04)* (pp. 415-422). doi:10.3115/1218955.1219008

Hassan, H., Hassan, A., & Emam, O. (2006). Unsupervised information extraction approach using graph mutual reinforcement. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)* (pp. 501-508). doi:10.3115/1610075.1610144

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions (ACL demo'04)*. doi:10.3115/1219044.1219066

Lappin, S., & Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, *20*(4), 535–561.

Qiu, L., Kan, M., & Chua, T. (2004). A public reference implementation of the rap anaphora resolution algorithm. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)* (pp. 291-294).

Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. New York, NY: McGraw-Hill, Inc.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. doi:10.1145/361219.361220

Shen, W., Wang, J., Luo, P., & Wang, M. (2012). A graph-based approach for ontology population with named entities. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)* (pp. 345–354). doi:10.1145/2396761.2396807

Shen, W., Wang, J., Luo, P., Wang, M., & Yao, C. (2011). Reactor: A framework for semantic relation extraction and tagging over enterprise data. *Proceedings of the 20th International Conference Companion on World Wide Web (WWW'11)* (pp. 121–122). doi:10.1145/1963192.1963254

Shinyama, Y., & Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL'06)* (pp. 304–311). doi:10.3115/1220835.1220874

Skounakis, M., Craven, M., & Ray, S. (2003). Hierarchical hidden markov models for information extraction. *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)* (pp. 427–433).

Surdeanu, M., & Ciaramita, M. (2007). Robust information extraction with perceptrons. *Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE'07)*.

Swampillai, K., & Stevenson, M. (2010). Intersentential relations in information extraction corpora. *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*.

Wang, J., Han, J., & Li, C. (2007). Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, *19*(8), 1042–1056. doi:10.1109/TKDE.2007.1043

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, *3*, 1083–1106.

Zhou, G., Zhang, M., Ji, D., & Zhu, Q. (2007). Tree kernel-based relation extraction with context-sensitive structured parse tree information. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL'07)* (pp. 728-736).

Zhu, J., Nie, Z., Liu, X., Zhang, B., & Wen, J.-R. (2009). StatSnowball: A statistical approach to extracting entity relationships. *Proceedings of the 18th International Conference on World Wide Web (WWW'09)* (pp. 101-110). doi:10.1145/1526709.1526724

ENDNOTES

1 http://opennlp.sourceforge.net/
2 https://www.csie.ntu.edu.tw/~cjlin/libsvm/
3 http://tartarus.org/martin/PorterStemmer/
4 http://aye.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html

Wei Shen received the BS degree from Beihang University, China, in 2009 and the PhD degree in Computer Science from Tsinghua University, China, in 2014. He is an assistant professor in the College of Computer and Control Engineering, Nankai University, China. His research interests include entity linking, knowledge base population, and text mining. He is a recipient of the Google PhD fellowship.

Jianyong Wang received the PhD degree in Computer Science in 1999 from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a professor at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He was an assistant professor at Peking University, and visited Simon Fraser University, University of Illinois at Urbana-Champaign, and University of Minnesota at Twin Cities before joining Tsinghua University in December 2004. His research interests mainly include data mining and web information management. He has coauthored more than 60 papers in some leading international conferences and some top international journals. He is serving or ever served as a PC member for some leading international conferences, such as SIGKDD, VLDB, ICDE, WWW, and an associate editor of IEEE TKDE. He received the 2009 and 2010 HP Labs Innovation Research award, the 2009 Okawa Foundation Research Grant (Japan), WWW'08 best posters award, the Year 2007 Program for New Century Excellent Talents in University, The Ministry of Education of China, the Year 2013 second-class Prize for Natural Sciences, China Computer Federation, and the Year 2013 second-class Prize for Natural Sciences, The Ministry of Education of China. He is a senior member of the IEEE and a member of the ACM.

Ping Luo is currently an Associate Professor at Institute of Computing Technology, Chinese Academy of Sciences. Before joining ICT, he worked as Senior Research Scientist at the Hewlett-Packard Labs. His general area of research is knowledge discovery and machine learning. He has published 30+ papers in some prestigious refereed journals and conference proceedings, such as IEEE Transactions on Information Theory, IEEE Transactions on Knowledge and Data Engineering, Journal of Parallel and Distributed Computing, ACM SIGKDD, ACM CIKM, IJCAI. He is the recipient of the Doctoral Dissertation Award, China Computer Federation (2009), the ACM CIKM Best Student Paper Award (2012).

Min Wang joined Visa as the Senior Vice President and head of Visa Research in May 2015. In her role, Wang leads the research on data analytics, security and the future of payments. Prior to Visa, Wang was part of Google Research where she was a Senior Staff Research Scientist and research manager focused on knowledge integration and inferencing at Google's headquarters in Mountain View, California. Before Google, Wang was Director of HP Labs China in Beijing, China, where she was also named an HP Distinguished Technologist. Wang also held a senior research role as the manager of the Unified Data Analytics Department at IBM's Thomas J. Watson Research Center in Hawthorne, New York. Wang has received several distinguished research awards for her work on data management. In 2009, Wang received the ACM SIGMOD Test of Time Award for her 1999 SIGMOD paper, "Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets." Wang received her PhD in Computer Science from Duke University and BS and MS degrees, both in Computer Science, from Tsinghua University, Beijing, China.