Discovering Relations among Named Entities from Large Corpora

Takaaki Hasegawa [*]
Cyberspace Laboratories
Nippon Telegraph and Telephone Corporation
1-1 Hikarinooka, Yokosuka,
Kanagawa 239-0847, Japan
hasegawa.takaaki@lab.ntt.co.jp

Satoshi Sekine and **Ralph Grishman**
Dept. of Computer Science
New York University
715 Broadway, 7th floor,
New York, NY 10003, U.S.A.
{sekine,grishman}@cs.nyu.edu

Abstract

Discovering the significant relations embedded in documents would be very useful not only for information retrieval but also for question answering and summarization. Prior methods for relation discovery, however, needed large annotated corpora which cost a great deal of time and effort. We propose an unsupervised method for relation discovery from large corpora. The key idea is clustering pairs of named entities according to the similarity of context words intervening between the named entities. Our experiments using one year of newspapers reveals not only that the relations among named entities could be detected with high recall and precision, but also that appropriate labels could be automatically provided for the relations.

1 Introduction

Although Internet search engines enable us to access a great deal of information, they cannot easily give us answers to complicated queries, such as “a list of recent mergers and acquisitions of companies” or “current leaders of nations from all over the world”. In order to find answers to these types of queries, we have to analyze relevant documents to collect the necessary information. If many relations such as “Company A merged with Company B” embedded in those documents could be gathered and structured automatically, it would be very useful not only for information retrieval but also for question answering and summarization. Information Extraction provides methods for extracting information such as particular events and relations between entities from text. However, it is domain dependent and it could not give answers to those types of queries from Web documents which include widely various domains.

Our goal is automatically discovering useful relations among arbitrary entities embedded in large text corpora. We defined a relation broadly as an affiliation, role, location, part-whole, social relationship and so on between a pair of entities. For example, if the sentence, “George Bush was inaugurated as the president of the United States.” exists in documents, the relation, “George Bush”*(PERSON)* is the “*President of*” the “United States” *(GPE*[1]*)*, should be extracted. In this paper, we propose an unsupervised method of discovering relations among various entities from large text corpora. Our method does not need the richly annotated corpora required for supervised learning — corpora which take great time and effort to prepare. It also does not need any instances of relations as initial seeds for weakly supervised learning. This is an advantage of our approach, since we cannot know in advance all the relations embedded in text. Instead, we only need a named entity (NE) tagger to focus on the named entities which should be the arguments of relations. Recently developed named entity taggers work quite well and are able to extract named entities from text at a practically useful level.

The rest of this paper is organized as follows. We discuss prior work and their limitations in section 2. We propose a new method of relation discovery in section 3. Then we describe experiments and evaluations in section 4 and 5, and discuss the approach in section 6. Finally, we conclude with future work.

2 Prior Work

The concept of relation extraction was introduced as part of the Template Element Task, one of the information extraction tasks in the Sixth Message Understanding Conference (MUC-6) (Defense Advanced Research Projects Agency, 1995). MUC-7 added a Template Relation Task, with three relations. Following MUC, the Automatic Content Extraction (ACE) meetings (National Institute of Standards and Technology, 2000) are pursuing informa-

[*] This work is supported by Nippon Telegraph and Telephone (NTT) Corporation’s one-year visiting program at New York University.

[1] GPE is an acronym introduced by the ACE program to represent a Geo-Political Entity — an entity with land and a government.

tion extraction. In the ACE Program[2], Relation Detection and Characterization (RDC) was introduced as a task in 2002. Most of approaches to the ACE RDC task involved supervised learning such as kernel methods (Zelenko et al., 2002) and need richly annotated corpora which are tagged with relation instances. The biggest problem with this approach is that it takes a great deal of time and effort to prepare annotated corpora large enough to apply supervised learning. In addition, the varieties of relations were limited to those defined by the ACE RDC task. In order to discover knowledge from diverse corpora, a broader range of relations would be necessary.

Some previous work adopted a weakly supervised learning approach. This approach has the advantage of not needing large tagged corpora. Brin proposed the bootstrapping method for relation discovery (Brin, 1998). Brin's method acquired patterns and examples by bootstrapping from a small initial set of seeds for a particular relation. Brin used a few samples of book titles and authors, collected common patterns from context including the samples and finally found new examples of book title and authors whose context matched the common patterns. Agichtein improved Brin's method by adopting the constraint of using a named entity tagger (Agichtein and Gravano, 2000). Ravichandran also explored a similar method for question answering (Ravichandran and Hovy, 2002). These approaches, however, need a small set of initial seeds. It is also unclear how initial seeds should be selected and how many seeds are required. Also their methods were only tried on *functional* relations, and this was an important constraint on their bootstrapping.

The variety of expressions conveying the same relation can be considered an example of paraphrases, and so some of the prior work on paraphrase acquisition is pertinent to relation discovery. Lin proposed another weakly supervised approach for discovering paraphrase (Lin and Pantel, 2001). Firstly Lin focused on verb phrases and their fillers as subject or object. Lin's idea was that two verb phrases which have similar fillers might be regarded as paraphrases. This approach, however, also needs a sample verb phrase as an initial seed in order to find similar verb phrases.

3 Relation Discovery

3.1 Overview

We propose a new approach to relation discovery from large text corpora. Our approach is based on context based clustering of pairs of entities. We assume that pairs of entities occurring in similar context can be clustered and that each pair in a cluster is an instance of the same relation. Relations between entities are discovered through this clustering process. In cases where the contexts linking a pair of entities express multiple relations, we expect that the pair of entities either would not be clustered at all, or would be placed in a cluster corresponding to its most frequently expressed relation, because its contexts would not be sufficiently similar to contexts for less frequent relations. We assume that useful relations will be frequently mentioned in large corpora. Conversely, relations mentioned once or twice are not likely to be important.

Our basic idea is as follows:

1. tagging named entities in text corpora
2. getting co-occurrence pairs of named entities and their context
3. measuring context similarities among pairs of named entities
4. making clusters of pairs of named entities
5. labeling each cluster of pairs of named entities

We show an example in Figure 1. First, we find the pair of *ORGANIZATION*s (*ORG*) *A* and *B*, and the pair of *ORGANIZATION*s (*ORG*) *C* and *D*, after we run the named entity tagger on our newspaper corpus. We collect all instances of the pair *A* and *B* occurring within a certain distance of one another. Then, we accumulate the context words intervening between *A* and *B*, such as "be offer to buy", "be negotiate to acquire".[3] In same way, we also accumulate context words intervening between *C* and *D*. If the set of contexts of *A* and *B* and those of *C* and *D* are similar, these two pairs are placed into the same cluster. *A* – *B* and *C* – *D* would be in the same relation, in this case, *merger and acquisition (M&A)*. That is, we could discover the relation between these *ORGANIZATION*s.

3.2 Named entity tagging

Our proposed method is fully unsupervised. We do not need richly annotated corpora or any initial manually selected seeds. Instead of them, we use a named entity (NE) tagger. Recently developed named entity taggers work quite well and extract named entities from text at a practically usable

[2] A research and evaluation program in information extraction organized by the U.S. Government.

[3] We collect the base forms of words which are stemmed by a POS tagger (Sekine, 2001). But verb past participles are distinguished from other verb forms in order to distinguish the passive voice from the active voice.

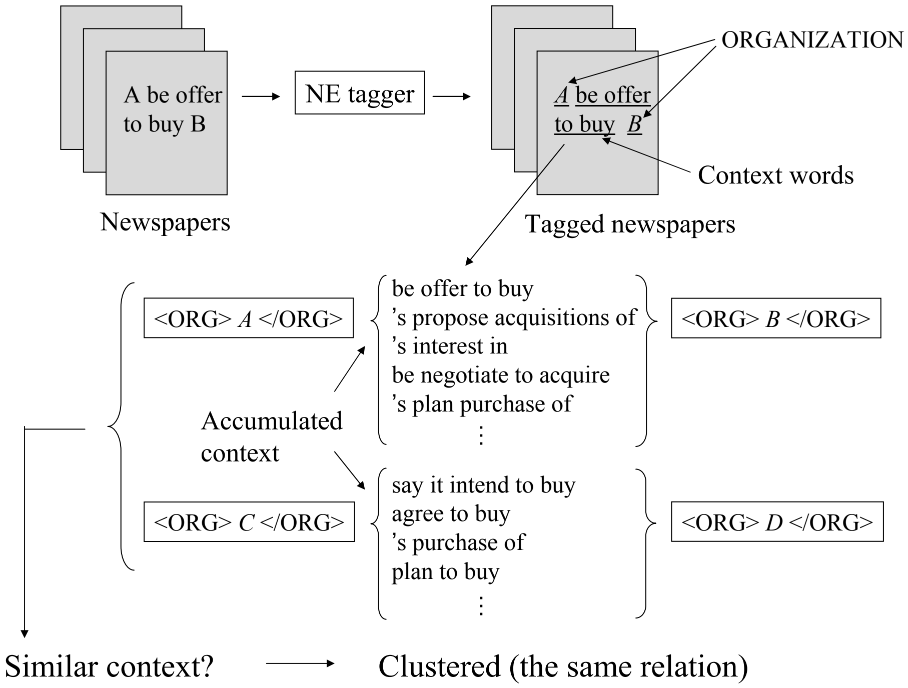

Figure 1: Overview of our basic idea

level. In addition, the set of types of named entities has been extended by several research groups. For example, Sekine proposed 150 types of named entities (Sekine et al., 2002). Extending the range of NE types would lead to more effective relation discovery. If the type *ORGANIZATION* could be divided into subtypes, *COMPANY*, *MILITARY*, *GOVERNMENT* and so on, the discovery procedure could detect more specific relations such as those between *COMPANY* and *COMPANY*.

We use an extended named entity tagger (Sekine, 2001) in order to detect useful relations between extended named entities.

3.3 NE pairs and context

We define the co-occurrence of NE pairs as follows: two named entities are considered to co-occur if they appear within the same sentence and are separated by at most *N* intervening words.

We collect the intervening words between two named entities for each co-occurrence. These words, which are stemmed, could be regarded as the context of the pair of named entities. Different orders of occurrence of the named entities are also considered as different contexts. For example, $e_1 ... e_2$ and $e_2 ... e_1$ are collected as different contexts, where e_1 and e_2 represent named entities.

Less frequent pairs of NEs should be eliminated because they might be less reliable in learning relations. So we have set a frequency threshold to remove those pairs.

3.4 Context similarity among NE pairs

We adopt a vector space model and cosine similarity in order to calculate the similarities between the set of contexts of NE pairs. We only compare NE pairs which have the same NE types, e.g., one *PERSON – GPE* pair and another *PERSON – GPE* pair. We define a *domain* as a pair of named entity types, e.g., the *PERSON-GPE* domain. For example, we have to detect relations between *PERSON* and *GPE* in the *PERSON-GPE* domain.

Before making context vectors, we eliminate stop words, words in parallel expressions, and expressions peculiar to particular source documents (examples of these are given below), because these expressions would introduce noise in calculating similarities.

A context vector for each NE pair consists of the bag of words formed from all intervening words from *all* co-occurrences of two named entities. Each word of a context vector is weighed by tf*idf, the product of term frequency and inverse document frequency. Term frequency is the number of occurrences of a word in the collected context words. The order of co-occurrence of the named entities is also considered. If a word w_i occurred L times in context $e_1 ... e_2$ and M times in context $e_2 ... e_1$, the term frequency tf_i of the word w_i is defined as $L - M$, where e_1 and e_2 are named entities. We think that this term frequency of a word in different orders would be effective to detect the direction of a relation if the arguments of a relation have the same NE types. Document frequency is the number of documents which include the word.

If the norm $|\alpha|$ of the context vector α is extremely small due to a lack of content words, the cosine similarity between the vector and others might be unreliable. So, we also define a norm threshold in advance to eliminate short context vectors.

The cosine similarity $cosine(\theta)$ between context vectors α and β is calculated by the following formula.

$$cosine(\theta) = \frac{\alpha \cdot \beta}{|\alpha||\beta|}$$

Cosine similarity varies from 1 to -1. A cosine similarity of 1 would mean these NE pairs have exactly the same context words with the NEs appearing predominantly in the same order, and a cosine similarity of -1 would mean these NE pairs have exactly the same context words with the NEs appearing predominantly in reverse order.

3.5 Clustering NE pairs

After we calculate the similarity among context vectors of NE pairs, we make clusters of NE pairs based on the similarity. We do not know how many clusters we should make in advance, so we adopt hierarchical clustering. Many clustering methods were proposed for hierarchical clustering, but we adopt complete linkage because it is conservative in making clusters. The distance between clusters is taken to be the distance of the furthest nodes between clusters in complete linkage.

3.6 Labeling clusters

If most of the NE pairs in the same cluster had words in common, the common words would represent the characterization of the cluster. In other words, we can regard the common words as the characterization of a particular relation.

We simply count the frequency of the common words in all combinations of the NE pairs in the same cluster. The frequencies are normalized by the number of combinations. The frequent common words in a cluster would become the label of the cluster, i.e. they would become the label of the relation, if the cluster would consist of the NE pairs in the same relation.

4 Experiments

We experimented with one year of The New York Times (1995) as our corpus to verify our proposed method. We determined three parameters for thresholds and identified the patterns for parallel expressions and expressions peculiar to The New York Times as ignorable context. We set the maximum context word length to 5 words and set the frequency threshold of co-occurring NE pairs to 30 empirically. We also used the patterns, “`,.*,`”, “`and`” and “`or`” for parallel expressions, and the pattern “`)  --`” (used in datelines at the beginning of articles) as peculiar to The New York Times. In our experiment, the norm threshold was set to 10. We also used stop words when context vectors are made. The stop words include symbols and words which occurred under 3 times as infrequent words and those which occurred over 100,000 times as highly frequent words.

We applied our proposed method to The New York Times 1995, identified the NE pairs satisfying our criteria, and extracted the NE pairs along with their intervening words as our data set. In order to evaluate the relations detected automatically, we analyzed the data set manually and identified the relations for two different domains. One was the *PERSON-GPE* (PER-GPE) domain. We obtained 177 distinct NE pairs and classified them into 38 classes (relations) manually. The other was the *COMPANY-COMPANY* (COM-COM) domain. We got 65 distinct NE pairs and classified them into 10 classes manually. However, the types of both arguments of a relation are the same in the COM-COM domain. So the COM-COM domain includes symmetrical relations as well as asymmetrical relations. For the latter, we have to distinguish the different orders of arguments. We show the types of classes and the number in each class in Table 1. The errors in NE tagging were eliminated to evaluate our method correctly.

5 Evaluation

We evaluated separately the placement of the NE pairs into clusters and the assignment of labels to these clusters. In the first step, we evaluated clusters consisting of two or more pairs. For each cluster, we determined the relation (*R*) of the cluster as the most frequently represented relation; we call this the *major relation* of the cluster. NE pairs with relation *R* in a cluster whose major relation was *R* were counted as correct; the correct pair count, $N_{correct}$, is defined as the total number of correct pairs in all clusters. Other NE pairs in the cluster were counted as incorrect; the incorrect pair count, $N_{incorrect}$, is also defined as the total number of incorrect pairs in all clusters. We evaluated clusters based on Recall, Precision and F-measure. We defined these mea-

PER-GPE	President	Senator	Governor	Prime Minister	Player	Living	Coach
# NE pairs	28	21	17	16	12	9	8
PER-GPE	Republican	Secretary	Mayor	Enemy	Working	others(2 and 3)	others(only 1)
# NE pairs	8	7	5	5	4	20	17
COM-COM	M&A	Rival	Parent	Alliance	Joint Venture	Trading	others(only 1)
# NE pairs	35	8	8	6	2	2	4

Table 1: Manually classified relations which are extracted from Newspapers

sures as follows.

Recall (R) How many correct pairs are detected out of all the key pairs? The key pair count, N_{key}, is defined as the total number of pairs manually classified in clusters of two or more pairs. Recall is defined as follows:

$$R = \frac{N_{correct}}{N_{key}}$$

Precision (P) How many correct pairs are detected among the pairs clustered automatically? Precision is defined as follows:

$$P = \frac{N_{correct}}{N_{correct} + N_{incorrect}}$$

F-measure (F) F-measure is defined as a combination of recall and precision according to the following formula:

$$F = \frac{2RP}{R + P}$$

These values vary depending on the threshold of cosine similarity. As the threshold is decreased, the clusters gradually merge, finally forming one big cluster. We show the results of complete linkage clustering for the *PERSON-GPE* (PER-GPE) domain in Figure 2 and for the *COMPANY-COMPANY* (COM-COM) domain in Figure 3. With these metrics, precision fell as the threshold of cosine similarity was lowered. Recall increased until the threshold was almost 0, at which point it fell because the total number of correct pairs in the remaining few big clusters decreased. The best F-measure was 82 in the PER-GPE domain, 77 in the COM-COM domain. In both domains, the best F-measure was found near 0 cosine similarity. Generally, it is difficult to determine the threshold of similarity in advance. Since the best threshold of cosine similarity was almost same in the two domains, we fixed the cosine threshold at a single value just above zero for both domains for simplicity.

We also investigated each cluster with the threshold of cosine similarity just above 0. We got 34

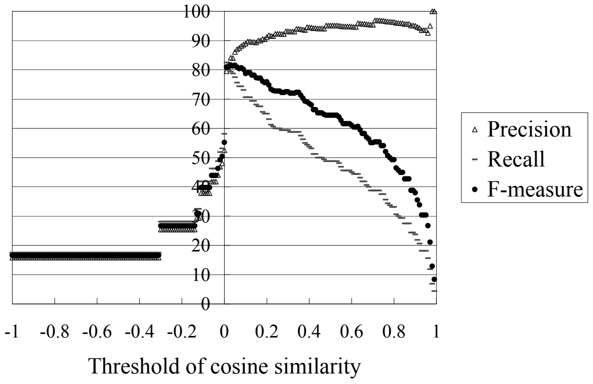

Figure 2: F-measure, recall and precision by varying the threshold of cosine similarity in complete linkage clustering for the *PERSON-GPE* domain

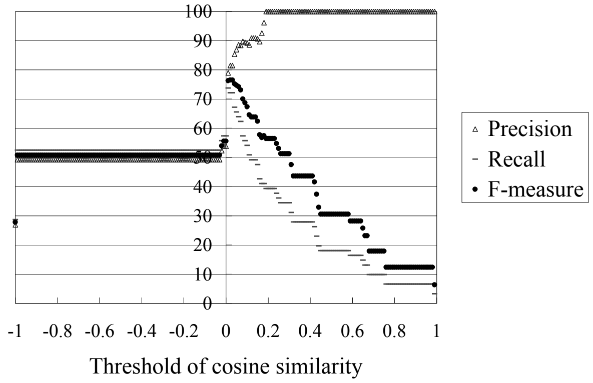

Figure 3: F-measure, recall and precision by varying the threshold of cosine similarity in complete linkage clustering for the *COMPANY-COMPANY* domain

	Precision	Recall	F-measure
PER-GPE	79	83	80
COM-COM	76	74	75

Table 2: F-measure, recall and precision with the threshold of cosine similarity just above 0

Major relations	Ratio	Common words (Relative frequency)
President	17 / 23	President (1.0), president (0.415), ...
Senator	19 / 21	Sen. (1.0), Republican (0.214), Democrat (0.133), republican (0.133), ...
Prime Minister	15 / 16	Minister (1.0), minister (0.875), Prime (0.875), prime (0.758), ...
Governor	15 / 16	Gov. (1.0), governor (0.458), Governor (0.3), ...
Secretary	6 / 7	Secretary (1.0), secretary (0.143), ...
Republican	5 / 6	Rep. (1.0), Republican (0.667), ...
Coach	5 / 5	coach (1.0), ...
M&A	10 / 11	buy (1.0), bid (0.382), offer (0.273), purchase (0.273), ...
M&A	9 / 9	acquire (1.0), acquisition (0.583), buy (0.583), agree (0.417), ...
Parent	7 / 7	parent (1.0), unit (0.476), own (0.143), ...
Alliance	3 / 4	join (1.0)

Table 3: Major relations in clusters and the most frequent common words in each cluster

PER-GPE clusters and 15 COM-COM clusters. We show the F-measure, recall and precision at this cosine threshold in both domains in Table 2. We got 80 F-measure in the PER-GPE domain and 75 F-measure in the COM-COM domain. These values were very close to the best F-measure.

Then, we evaluated the labeling of clusters of NE pairs. We show the larger clusters for each domain, along with the ratio of the number of pairs bearing the major relation to the total number of pairs in each cluster, on the left in Table 3. (As noted above, the major relation is the most frequently represented relation in the cluster.) We also show the most frequent common words and their relative frequency in each cluster on the right in Table 3. If two NE pairs in a cluster share a particular context word, we consider these pairs to be linked (with respect to this word). The relative frequency for a word is the number of such links, relative to the maximal possible number of links ($N(N-1)/2$ for a cluster of N pairs). If the relative frequency is 1.0, the word is shared by all NE pairs. Although we obtained some meaningful relations in small clusters, we have omitted the small clusters because the common words in such small clusters might be unreliable. We found that all large clusters had appropriate relations and that the common words which occurred frequently in those clusters accurately represented the relations. In other words, the frequent common words could be regarded as suitable labels for the relations.

6 Discussion

The results of our experiments revealed good performance. The performance was a little higher in the PER-GPE domain than in the COM-COM domain, perhaps because there were more NE pairs with high cosine similarity in the PER-GPE domain than in the COM-COM domain. However, the graphs in both domains were similar, in particular when the cosine similarity was under 0.2.

We would like to discuss the differences between the two domains and the following aspects of our unsupervised method for discovering the relations:

- properties of relations
- appropriate context word length
- selecting best clustering method
- covering less frequent pairs

We address each of these points in turn.

6.1 Properties of relations

We found that the COM-COM domain was more difficult to judge than the PER-GPE domain due to the similarities of relations. For example, the pair of companies in *M&A* relation might also subsequently appear in the *parent* relation.

Asymmetric properties caused additional difficulties in the COM-COM domain, because most relations have directions. We have to recognize the direction of relations, $A \rightarrow B$ vs. $B \rightarrow A$, to distinguish, for example, “A is parent company of B” and “B is parent company of A”. In determining the similarities between the NE pairs *A* and *B* and the NE pairs *C* and *D*, we must calculate both the similarity $A \rightarrow B$ with $C \rightarrow D$ and the similarity $A \rightarrow B$ with $D \rightarrow C$. Sometimes the wrong correspondence ends up being favored. This kind of error was observed in 2 out of the 15 clusters, due to the fact that words happened to be shared by NE pairs aligned in the wrong direction more than in right direction.

6.2 Context word length

The main reason for undetected or mis-clustered NE pairs in both domains is the absence of common words in the pairs’ context which explicitly represent the particular relations. Mis-clustered NE pairs were clustered based on another common word which occurred by accident. If the maximum context length were longer than the limit of 5 words which we set in the experiments, we could detect additional common words, but the noise would also increase. In our experiments, we used only the words between the two NEs. Although the outer context words (preceding the first NE or following the second NE) may be helpful, extending the context in this way will have to be carefully evaluated. It is future work to determine the best context word length.

6.3 Clustering method

We tried single linkage and average linkage as well as complete linkage for making clusters. Complete linkage was the best clustering method because it yielded the highest F-measure. Furthermore, for the other two clustering methods, the threshold of cosine similarity producing the best F-measure was different in the two domains. In contrast, for complete linkage the optimal threshold was almost the same in the two domains. The best threshold of cosine similarity in complete linkage was determined to be just above 0; when this threshold reaches 0, the F-measure drops suddenly because the pairs need not share any words. A threshold just above 0 means that each combination of NE pairs in the same cluster shares at least one word in common — and most of these common words were pertinent to the relations. We consider that this is relevant to context word length. We used a relatively small maximum context word length – 5 words – making it less likely that noise words appear in common across different relations. The combination of complete linkage and small context word length proved useful for relation discovery.

6.4 Less frequent pairs

As we set the frequency threshold of NE co-occurrence to 30, we will miss the less frequent NE pairs. Some of those pairs might be in valuable relations. For the less frequent NE pairs, since the context varieties would be small and the norms of context vectors would be too short, it is difficult to reliably classify the relation based on those pairs. One way of addressing this defect would be through bootstrapping. The problem of bootstrapping is how to select initial seeds; we could resolve this problem with our proposed method. NE pairs which have many context words in common in each cluster could be promising seeds. Once these seeds have been established, additional, lower-frequency NE pairs could be added to these clusters based on more relaxed keyword-overlap criteria.

7 Conclusion

We proposed an unsupervised method for relation discovery from large corpora. The key idea was clustering of pairs of named entities according to the similarity of the context words intervening between the named entities. The experiments using one year's newspapers revealed not only that the relations among named entities could be detected with high recall and precision, but also that appropriate labels could be automatically provided to the relations. In the future, we are planning to discover less frequent pairs of named entities by combining our method with bootstrapping as well as to improve our method by tuning parameters.

8 Acknowledgments

This research was supported in part by the Defense Advanced Research Projects Agency as part of the Translingual Information Detection, Extraction and Summarization (TIDES) program, under Grant N66001-001-1-8917 from the Space and Naval Warfare Systems Center, San Diego, and by the National Science Foundation under Grant ITS-00325657. This paper does not necessarily reflect the position of the U.S. Government.

We would like to thank Dr. Yoshihiko Hayashi at Nippon Telegraph and Telephone Corporation, currently at Osaka University, who gave one of us (T.H.) an opportunity to conduct this research.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proc. of the 5th ACM International Conference on Digital Libraries (ACM DL'00)*, pages 85–94.

Sergey Brin. 1998. Extracting patterns and relations from world wide web. In *Proc. of WebDB Workshop at 6th International Conference on Extending Database Technology (WebDB'98)*, pages 172–183.

Defense Advanced Research Projects Agency. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann Publishers, Inc.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pages 323–328.

National Institute of Standards and Technology. 2000. Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/index.htm.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 41–47.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proc. of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1818–1824.

Satoshi Sekine. 2001. OAK System (English Sentence Analyzer). http://nlp.cs.nyu.edu/oak/.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 71–78.