Automatic Paraphrase Acquisition from News Articles

Yusuke Shinyama
Department of Computer Science
New York University
715 Broadway, 7th floor, New York, NY, 10003
yusuke@cs.nyu.edu

Satoshi Sekine
Department of Computer Science
New York University
715 Broadway, 7th floor, New York, NY, 10003
sekine@cs.nyu.edu

Kiyoshi Sudo
Department of Computer Science
New York University
715 Broadway, 7th floor, New York, NY, 10003
sudo@cs.nyu.edu

ABSTRACT

Paraphrases play an important role in the variety and complexity of natural language documents. However, they add to the difficulty of natural language processing. Here we describe a procedure for obtaining paraphrases from news articles. Articles derived from different newspapers can contain paraphrases if they report the same event on the same day. We exploit this feature by using Named Entity recognition. Our approach is based on the assumption that Named Entities are preserved across paraphrases. We applied our method to articles of two domains and obtained notable examples. Although this is our initial attempt at automatically extracting paraphrases from a corpus, the results are promising.

1. INTRODUCTION

Expressing one thing in other words, or "paraphrasing", plays an important role in the variety and complexity of natural language documents. One can express a single event in thousands of ways in natural language sentences. A creative writer uses lots of paraphrases to state a single fact. This greatly adds to the difficulty of natural language processing.

Table 1 shows how the headlines differ in several newspapers. Although every expression reports the same event – Bush's decision for government funding for people in New York – each expression differs considerably from the others.

Many natural language applications, such as Information Retrieval, Machine Translation, Question Answering, Text Summarization, or Information Extraction, need to handle these expressions correctly. Because analyzing these expressions at semantic level is a rather difficult task, we hope to build a paraphrase database to find expressions which have the same meaning. However, building such databases by hand is still difficult. There are two reasons: the first reason is that there are too many possible language expressions for someone to come up with. Even if several people work on this task, it is still laborious to cover many common expressions. The second reason is that expressions considered as paraphrases are different from domain to domain. Even if two expressions can be regarded as having the same meaning in a certain domain, it is not possible to generalize them to other domains.

So we are trying to create a system that automatically acquires paraphrases from given corpora of a specific domain. Even though their usage is limited to a certain domain, it is still useful for many applications. In this paper, we describe an approach to automatic paraphrase acquisition from corpora. Our main focus is Information Extraction (IE). In an IE application, a system uses patterns to capture events which are relevant to a certain domain. Although there have been several efforts to obtain such patterns automatically, little work has addressed the problem of capturing the semantic knowledge of such patterns, which is crucial for IE. Using a paraphrase database, we can connect one pattern to another. We expect this will reduce the cost of creating IE knowledge by hand. Although our approach aims to collect paraphrases for IE applications, our method can be applied to other purposes also.

2. CHALLENGES

To acquire paraphrases automatically, we focused on news articles that describe the same event. Take a look at the examples in Table 1. These headlines are taken from several news articles on the same day. If we can find these articles in different newspapers on a certain day, it is likely that they contain similar expressions; i.e. paraphrases.

However, the difficulty of paraphrase acquisition is to recognize that one sentence has the same meaning as another. These expressions may differ from the others not only in lexical properties, but also in syntactic features. By looking at Table 1, one can easily observe that a simple criterion is not enough to find paraphrases.

Our basic concept is to use Named Entity (NE) to find such expressions reporting the same event. NE is a proper expression such as names of organizations, persons, locations, dates, or numerical expressions [2]. In Table 1, "Bush", "New York" and "$20 billion" are regarded as NEs. Since they are indispensable to report an event, NEs are often preserved across different newspapers. Therefore we can expect that if two sentences share several comparable NEs, it is likely that they are reporting the same event. This likelihood increases as the number of NEs shared by two sentences increases. Here, using NE recognition techniques, headlines 2. and 3. can be generalized as follows:

- Bush, in New York, Affirms $20 Billion Aid Pledge
  $\Rightarrow$ *PERSON*$_1$, in *LOCATION*$_1$, affirms *MONEY*$_1$ Aid Pledge
- Bush Reassures New York of $20 Billion
  $\Rightarrow$ *PERSON*$_1$ Reassures *LOCATION*$_1$ of *MONEY*$_1$

This way, we can find the comparable expressions, or paraphrases from corpora by using NEs. So far we have applied our method to

.

No	Newspaper	Headline
1.	CNN	Bush says he'll deliver $20 billion to NY
2.	New York Times	Bush, in New York, Affirms $20 Billion Aid Pledge
3.	Washington Post	Bush Reassures New York of $20 Billion

Table 1: Expressions of the same event

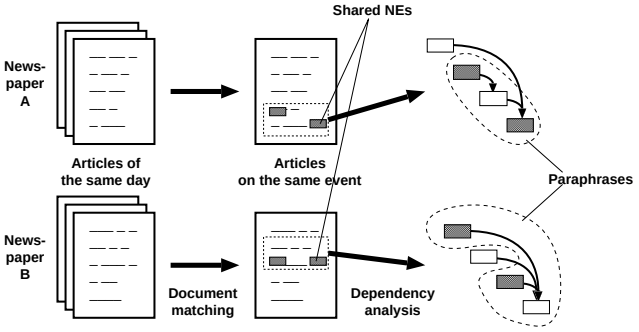


Figure 1: Overall method of paraphrase acquisition

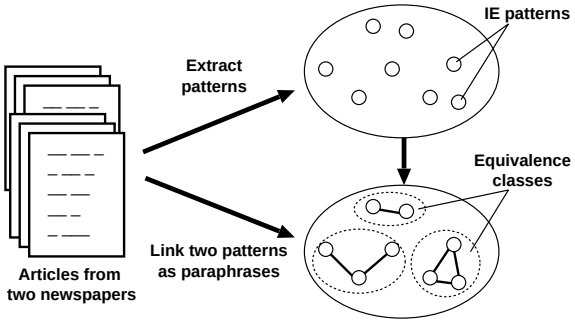


Figure 3: Actual experiment using IE patterns

two domains in Japanese newspapers and obtained some notable examples.

There are a few approaches for obtaining paraphrases automatically. Barzilay et al. used parallel translations derived from one original document [1]. They targeted literary works and used word alignment techniques developed for MT. However the syntactic variety of the resultant expressions is limited since they used only part-of-speech tags to identify the syntactic properties. In addition, compared with our method using newspapers, their resources are relatively scarce. Torisawa et al. proposed a learning method for automatic paraphrasing of Japanese noun phrases [6]. But this is also limited to a certain type of noun phrases.

3. ALGORITHM

3.1 Overview

As we stated in the previous section, our approach is based on the following assumption: NEs are preserved across paraphrases. So if the portions of each sentence in the articles share several comparable NEs, they are likely to be expressing the same meaning; in other words, they are paraphrases. The expectation increases as the number of NEs shared by the portions increases.

Paraphrase acquisition goes as follows. First we find articles in a certain domain from two newspapers. We use an existing IR system to obtain articles from a given class of events, such as murders or personnel affairs. Then we find pairs of articles which report the same event. In this stage we use a TF/IDF based method developed for Topic Detection and Tracking (TDT). Next we compare all the sentences in each article to find pairs of sentences sharing comparable NEs. Then we extract appropriate portions of sentences using a dependency tree. A dependency tree can be used later to reconstruct a original phrase. If the number of comparable NEs which both portions contain exceeds a certain threshold, we adopt them as paraphrases. Finally we generalize an NE as a variable in retrieved phrases so that these phrases can be applied to other sentences. The overall process is illustrated in Figure 1.

Additionally, we need to consider the domain of the expressions. Otherwise our method yields a lot of noise. For example, two expressions "*Bush has expressed his confidence in Koizumi's reforms*" and "*Bush and Koizumi watched a demonstration of horseback archery*" are both found in the articles from the same day and both contain the comparable NEs (*Bush* and *Koizumi*), but they are not paraphrases. So we try to filter out such noise using a set of IE patterns obtained from the same articles in advance. In this way we can limit our patterns to only those concerning a certain domain.

Sudo et al. described a procedure for automatically gathering common patterns appearing frequently in a set of articles about a given topic [5]. Each IE pattern has slots which can be filled by NEs. For example, the sentence "*Vice President Osamu Kuroda of Nihon Yamamura Glass Corp. was promoted to President.*" contains four patterns found for the personnel domain, as shown in Figure 2. NEs in these patterns are generalized into slots which hold the types of the NEs and the case roles of each node are preserved. We apply these obtained patterns to the articles itself, and then find paraphrases only among those which match any of the patterns. This means we find paraphrases among these IE patterns. Actually this is done by linking two IE patterns as paraphrases. These links form a set of equivalence classes, in which each pattern conveys the same meaning (See Figure 3).

3.2 Details

Now we describe the details. Our method can be divided into 4 stages:

1) Preprocessing articles

First we obtain pairs of articles of a certain domain from two newspapers, as a source of both IE patterns and paraphrases. First we obtain relevant articles for a domain from one newspaper, and then we find articles which report the same event from the other newspaper. In this experiment, we used a stochastic-based IR system by Murata et al. [3] to get articles of a specified domain. We pick up the most relevant 300 articles for a domain. For each relevant article from one of the newspapers, we search for an article corresponding to the first article from the other newspaper. This is done by calculating the similarity between two articles and taking the one whose similarity is the best. Since this task is very similar to the task defined in TDT [8], we used a technique developed in TDT. We implemented this part based on the University of Massachusetts system, which worked the best for our purpose [4]. The similarity $S_a(a_1, a_2)$ of two articles a_1 and a_2 is defined as follows:

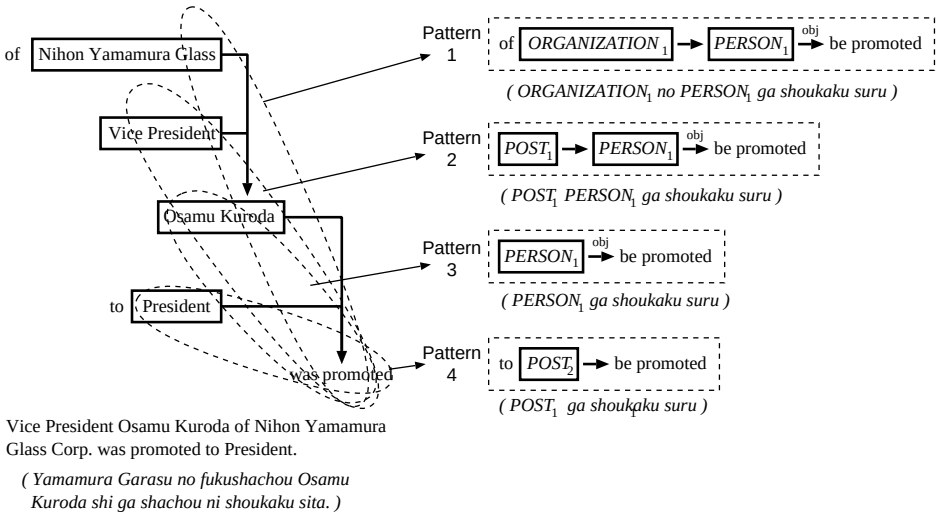


Figure 2: IE Pattern Extraction

$$S_a(a_1, a_2) = \cos(W_1, W_2) \tag{1}$$

$$W^i = TF(w_i) * IDF(w_i) \tag{2}$$

$$TF(w_i) = \frac{f(w_i)}{f(w_i) + 0.5 + 1.5 * \frac{dl}{avgdl}} \tag{3}$$

$$IDF(w_i) = \frac{\log(\frac{C+0.5}{df(w_i)})}{\log(C+1)} \tag{4}$$

Here W_1 and W_2 are vectors with elements W_1^i and W_2^i for article a_1 and a_2, with dimension equal to the number of NEs in the corpus. $f(w_i)$ is the number of times NE w_i occurs in the article. $df(w_i)$ is the document (article) frequency, which is the number of articles containing the NE w_i. dl is the document length. C is the number of articles, and $avgdl$ is the average article length.

We apply this metric to the NEs appearing in an article and adopt the article pairs whose similarity is above a certain threshold. In this stage we use a simple dictionary-based NE tagging system to pick up NEs, instead of the one used in later stage. This system picks up only words which are not contained in a common noun dictionary and doesn't recognize the type of NEs.

2) Acquiring IE patterns

Next we run the IE pattern acquisition system for the pairs of articles [5]. This system performs NE tagging and dependency analysis, and picks several paths of nodes in a dependency tree as IE patterns. In this experiment, we use IE patterns which appear more then once in the corpus and contain at lease one NE.

3) Preprocessing sentences

Now we take a closer look at a pair of articles which report the same event. We mark all NEs using an statistical NE tagging system [7]. Next we apply a dependency analyzer to the sentences. Here Juman and KNP were used as the morphological analyzer and dependency analyzer respectively. Thus we have a set of NE-tagged dependency trees for each article. Here we apply the obtained IE patterns to the sentences. We drop a sentence that doesn't match any of the patterns. For sentences which do match one or more patterns, an instance of each pattern is created and attached to the sentence. The variables in these patterns are filled with the actual NEs.

This stage is illustrated in Figure 4. Suppose sentences A and B contain paraphrases. Sentence A matches pattern 1 and sentence B matches pattern 2. These patterns are attached to the sentences and each slot in the patterns is filled with the actual NE (here, $POST_1$ slot is filled with the actual NE *"President"*).

4) Extracting paraphrases

Now we can get paraphrases. First we take pairs of similar sentences. To penalize frequently occurring NEs, this is done by calculating TF/IDF based similarity in terms of comparable NEs for all possible pairs of sentences. Sentence similarity $S_s(s_1, s_2)$ of sentence s_1 and s_2 is defined as follows:

$$S_s(s_1, s_2) = \cos(W_1, W_2) \tag{5}$$

$$W^i = TF(w_i) * IDF(w_i) \tag{6}$$

$$TF(w_i) = f(w_i) \tag{7}$$

$$IDF(w_i) = \log(\frac{C}{df(w_i)}) \tag{8}$$

Here W_1 and W_2 are vectors with elements W_1^i and W_2^i for article s_1 and s_2. $f(w_i)$ is the number of NEs which are comparable to w_i in the sentence. $df(w_i)$ is the number of sentences in the article which contain NEs that is comparable to w_i. C is the number of NEs in the article.

We use substring matching to compare two NEs. This is because several NEs referring to one entity can take various forms, such as "Bush", "George W. Bush", or "Mr. Bush". Since we use Japanese newspapers for this experiment, we regard two NEs as comparable if one begins with the half of the beginning string of the other. [1]

Then we take pairs of sentences whose similarity is above a certain threshold. If two IE pattern attached to the two sentences share

[1] In Japanese, a name of a person can take the following forms: "Koizumi", "Koizumi Jun'ichirou", "Koizumi-san" etc.

Arrest Events:

Description	Hiring and firing of executives
Narrative	Domestic or international articles about hiring and firing of executives. Chairman, President, Director, CEO, COO, CFO or equivalent positions are targeted.

Personnel Affairs:

Description	Arresting robbery suspects
Narrative	Articles reporting arrest of robbery suspects or criminals. Multiple crimes such as murder accompanied by robbery or prior crimes of robbers should be included.

Table 2: Query Used for Article Retrieval

Newspaper	Mainichi	Nikkei
Articles	111373	181086

Table 3: Articles Used for the Experiment

the same number of comparable NEs, we link the two patterns as paraphrases.

In Figure 4, each sentence in the pair shares four comparable NEs (*"Nihon Yamamura Glass"*, *"President"*, *"Vice President"*, and *"Osamu Kuroda"*). Moreover, the variables in pattern 1 and 2 also have the same type ($POST_1$) and content (*"President"*). So we can conclude these two patterns are paraphrases.

4. EXPERIMENTS

We used one year of two Japanese newspapers (Mainichi and Nikkei) in this experiment. First we obtained the most relevant 300 articles from Mainichi newspaper (total of 111373 articles) for two domains, arrest events and personnel affairs (hiring and firing of executives). The descriptions and narratives we gave to the IR system are shown in Table 2. Next we find the corresponding articles of Nikkei newspaper from 181086 articles (See Table 3). The pairs whose similarity is below a certain threshold were dropped at this time. We got 294 pairs of articles in arrest events, and 289 pairs of articles in personnel affairs. Next we ran an IE pattern acquisition system for those articles. After dropping the patterns which appear only once, we got 725 patterns and 157 patterns respectively. Then we ran the paraphrase acquisition system for each pair or articles, and finally got total 136 pairs of paraphrases (a link between two IE patterns). The number of article pairs, obtained IE patterns and obtained paraphrases pairs are shown in Table 4.

5. EVALUATION

We evaluated our results in two respects: precision and coverage. To measure them, first we need to prepare the answer data. This is done by manually classifying the IE patterns for each domain. The criteria of classification are the following:

1. Do they describe the same event?

Domain	Arrest events	Personnel affairs
Article pairs	294	289
Sentences	4445	5962
Obtained IE patterns	725	157
Obtained paraphrase links	53	83

Table 4: Article pairs and IE patterns

Domain	Arrest events	Personnel affairs
IE patterns (forms a cluster)	363	129
Clusters	111	20

Table 5: Manually Classified Patterns

Domain	Arrest events	Personnel affairs
Obtained links (yield)	53	83
Correct links	26	78
Precision	49%	94%

Table 6: Precision of paraphrase links

2. If we use them in an actual IE application, do they capture the same information?

For example, the following two patterns are regarded as the same class [2] :

- *ORGANIZATION*$_1$ decides ϕ.
  (*ORGANIZATION*$_1$ *ha kettei suru.*)
- *ORGANIZATION*$_1$ confirms ϕ.
  (*ORGANIZATION*$_1$ *ha katameru.*)

In the above example, both describe the same event (decision) and the information captured by them is the same (which organization decides).

However, the following two patterns are not in the same class:

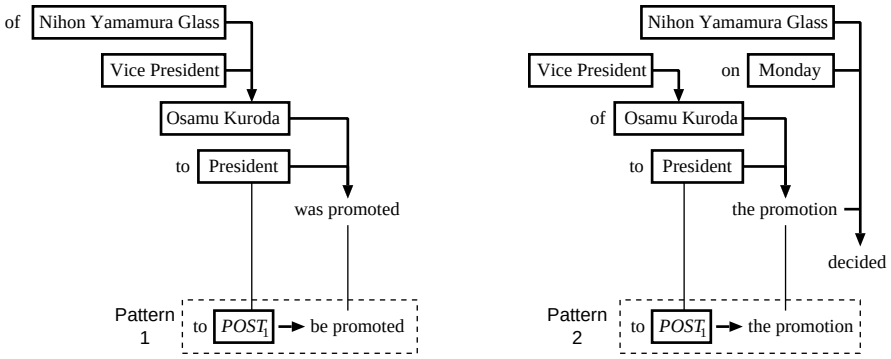
- ϕ is promoted to $POST_1$.
  ($POST_1$ *ni shoukaku suru*)
- $POST_1$ is promoted.
  ($POST_1$ *ha shoukaku suru*)

Although these patterns describe the same event (promotion), the information they capture is different. The former pattern captures the new post someone is promoted *to*. However the latter captures the old post someone is promoted *from*. So we put these patterns in different classes. This way, we get several clusters of patterns for each domain. We take only patterns which form a cluster and drop single-element patterns which do not have a paraphrase among the patterns. The result of manual classification in this experiment is shown in Table 5. We got 111 distinct clusters for arrest events and 20 for personnel affairs.

Now we describe how to measure precision and coverage. If the two patterns linked by our procedure are both in the same cluster, it is correct; otherwise, it is incorrect. Thus we measured the precision by counting how many paraphrase links are correct. The results are shown in Table 6. In arrest events domain, we got correct 26 out of 53 links and the precision was 49%. In the personnel affairs domain, we got correct 78 out of 83 links and the precision was 94%. We got quite high precision in personnel affairs, although it is not so high in the arrest domain. We will discuss the difference in the later section. Some examples of obtained correct and incorrect paraphrases are shown in Figure 6.

Next we define the coverage, how well the system obtains all the necessary links. This is done by calculating how many additional links are necessary to connect all the patterns in every cluster. See Figure 5. In this figure, cluster 1 has four obtained links. But the

[2]Note that these patterns are originally written in Japanese and include zero pronouns, which are shown as ϕ.



A: Vice President Osamu Kuroda of Nihon Yamamura Glass Corp. was promoted to President.

B: Nihon Yamamura Glass Corp., decided the promotion of Vice President Osamu Kuroda to President on Monday.

Figure 4: Sample Paraphrase Extraction

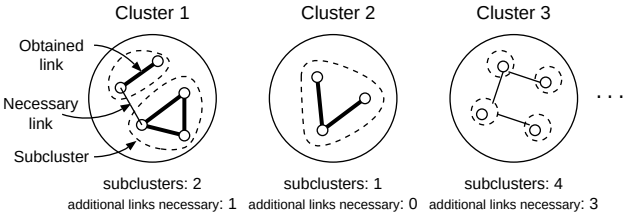


Figure 5: Evaluation of Coverage

Domain	Arrest events	Personnel affairs
Clusters Obtained	6	5
Additional links necessary L	230	57
Total necessary links M	252	109
Coverage	9%	47%

Table 7: Coverage over paraphrase links

cluster is split into two subclusters. So we need at least one additional link to unify these subclusters. Therefore, the number of additional links necessary in cluster 1 is 1. In cluster 2, all the patterns form one cluster, so no further link is needed. In cluster 3, there are four unconnected subclusters. So we need at least three additional links to unify these subclusters. In this way we can calculate the total number of additional links necessary L as:

$$L = \sum^{n} (s_i - 1) \tag{9}$$

Here s_i is the number of subclusters in cluster i. n is the number of clusters.

The smaller the value of L, the more coverage we get, which means the clusters obtained are properly formed. To normalize this value, we use the total number of the necessary links M to make the manually created clusters. This is calculated by summing the number of the links necessary to connect all the patterns in each cluster. So the definition of the coverage C is:

$$C = 1 - \frac{L}{M} \tag{10}$$

Here M is calculated as

$$M = \sum^{n} (p_i - 1) \tag{11}$$

where p_i is the number of the patterns in cluster i.

The results are shown in Table 7. In the arrest domain, links were discovered in 6 of the 111 clusters. 230 additional links would be needed to connect the patterns within all the clusters. The coverage in the arrest domain was 9%, which is not high and we will also discuss this problem in the next section. In the personnel affairs domain, links were discovered in 5 of the 20 clusters. 57 additional links would be needed to connect the patterns within all the clusters. The coverage in the personnel affairs domain was 47%.

6. DISCUSSION

Although this is our initial attempt at automatically extracting paraphrases from a corpus, the results are promising. In particular, we obtained expressions which are different in structure, such as "ϕ was promoted to *POST*$_1$" and "the promotion to *POST*$_1$ was decided". We also obtained expressions which can be regarded as paraphrases not in general, but in this particular domain. For example, *to admit (mitomeru)* and *to testify (kyoujutsu suru)* are generally not regarded as synonyms. But this semantic relationship is quite useful in this domain.

However, many problems remain. We reviewed our results in terms of the precision and the coverage:

Precision

Currently the precision in arrest events is not high. The main reason is that the average number of NEs in arrest events is low. This makes the obtained IE patterns short. Generally the more NEs contained in a pair of patterns, the more likely that they are paraphrases. However, only 41 patterns out of 725 patterns in arrest events domain contained two or more NEs. Additionally, the expressions used in this domain vary widely in meaning. This makes the obtained IE patterns equally varied. For example, there are 206 patterns in arrest events that contain only one *PERSON* NE. These expressions have varied predicates like *murder*, *die*, *run*, *abduct*, *rob*, *testify*, and so on. Since our method only considers the NEs contained in these patterns, a wrong pair of patterns can be paired as paraphrases in this domain.

The lack of NEs raises another problem in the calculation of the sentence similarity. Since we use only NEs for the calculation currently, sentences which contains fewer NEs are likely to be misidentified. So it is important to consider other words in calculating sentence similarity. A possible solution for this problem is to

Arrest events
Correct:

- *ORGANIZATION*$_1$ arrests ϕ.
  (ORGANIZATION$_1$ *ha taiho suru.)*
- the investigation authority of *ORGANIZATION*$_1$ arrests ϕ.
  (ORGANIZATION$_1$ *sousa toukyoku ha taiho suru.)*
- *PERSON*$_1$ admits ϕ.
  (PERSON$_1$ *ha mitomeru.)*
- *PERSON*$_1$ testifies ϕ.
  (PERSON$_1$ *ha kyoujutsu suru.)*

Incorrect:

- *PERSON*$_1$ is arrested.
  (PERSON$_1$ *ha taiho sareru.)*
- *PERSON*$_1$ conspires.
  (PERSON$_1$ *ha kyoubou suru.)*

Personnel affairs
Correct:

- ϕ is promoted to *POST*$_1$.
  (POST$_1$ *ni shoukaku suru.)*
- the promotion to *POST*$_1$ is decided.
  (POST$_1$ *no shoukaku wo kettei suru.)*
- *ORGANIZATION*$_1$ decides ϕ.
  (ORGANIZATION$_1$ *ha kettei suru.)*
- *ORGANIZATION*$_1$ confirms ϕ.
  (ORGANIZATION$_1$ *ha katameru.)*

Incorrect:

- *PERSON*$_1$ is promoted.
  (PERSON$_1$ *ha shoukaku suru.)*
- *PERSON*$_1$ hold successively ϕ.
  (PERSON$_1$ *ha rekinin suru.)*

Figure 6: Examples of obtained paraphrases
(Note that these patterns are originally written in Japanese and include zero pronouns.)

use not only NEs but also common nouns to find similar sentences.

Coverage

In this experiment, the coverage of obtained paraphrases is still low. However, we can expect that we will finally obtain a sufficient number of paraphrases, because the variety of paraphrases in a certain domain can saturate as we use a sufficient number of articles. Instead, the number of potentially obtainable paraphrases is more important because we want to be able to capture as wide a range of paraphrases as possible. So the problem is how to create a system that can handle such varied phrases. Our current IE patterns are limited to a single path in a dependency tree because of the limitation of the IE pattern extraction system we used [5]. For example, we cannot obtain a pattern like "*PERSON*$_1$ is promoted to *POST*$_1$", since the dependency tree of this expression has two branches. Now we are independently trying to extend them to include several branches to represent more complicated patterns, which would enable us to obtain more varied paraphrases.

Another possible problem is that not all sentences can be cleanly divided. A phrase used in one sentence may have inherently composite meanings and describe two events at once, whereas the expressions of the two events are separated in the other sentence. These patterns may reduce the overall coverage. For example, a pattern "*PERSON*$_1$ strangle ϕ" can be regarded as reporting two events: throttling and killing. This is one aspect of our future work.

Moreover, it is natural that comparable NEs appear in several forms which cannot be covered by current NE matching method, like "New York City", "NYC", or "the city". To solve this problem, we may need to consider co-reference information also. We are planning to refine the NE matching method in future.

7. ACKNOWLEDGMENTS

This research is supported 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 IIS-0081962. This paper does not necessarily reflect the position or the policy of the U.S. Government.

8. ADDITIONAL AUTHORS

Additional authors: Ralph Grishman
email: grishman@cs.nyu.edu

9. REFERENCES

[1] R. Barzilay and K. R. McKeown. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the ACL/EACL*, 2001.

[2] R. Grishman and B. Sundheim. Message Understanding Conference - 6: A Brief History. In *Proceedings of the COLING*, 1996.

[3] M. Murata, K. Uchimoto, H. Ozaku, and Q. Ma. Information Retrieval Based on Stochastic Models in IREX. In *Proceedings of the IREX Workshop*, 1994.

[4] R. Papka, J. Allen, and V. Lavrenko. UMASS Approaches to Detection and Tracking at TDT2. In *DARPA: Broadcast News Workshop*, 1999.

[5] K. Sudo and S. Sekine. Automatic Pattern Acquisition for Japanese Information Extraction. In *Proceedings of the HLT*, 2001.

[6] K. Torisawa. A Nearly Unsupervised Learning Method for Automatic Paraphrasing of Japanese Noun Phrases. In *Proceedings of the Sixth Natural Language Processing Pacific Rim*, 2001.

[7] K. Uchimoto, M. Murata, Q. Ma, H. Ozaku, and H. Isahara. Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 326–335, 2000.

[8] C. L. Wayne. Topic Detection & Tracking: A Case Study in Corpus Creation & Evaluation Methodologies. In *Proceedings of the LREC*, 1998.