Paraphrase Acquisition for Information Extraction

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

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

Abstract

We are trying to find paraphrases from Japanese news articles which can be used for Information Extraction. We focused on the fact that a single event can be reported in more than one article in different ways. However, certain kinds of noun phrases such as names, dates and numbers behave as "anchors" which are unlikely to change across articles. Our key idea is to identify these anchors among comparable articles and extract portions of expressions which share the anchors. This way we can extract expressions which convey the same information. Obtained paraphrases are generalized as templates and stored for future use.

In this paper, first we describe our basic idea of paraphrase acquisition. Our method is divided into roughly four steps, each of which is explained in turn. Then we illustrate several issues which we encounter in real texts. To solve these problems, we introduce two techniques: coreference resolution and structural restriction of possible portions of expressions. Finally we discuss the experimental results and conclusions.

1 Introduction

We are trying to obtain paraphrases which can be used for Information Extraction (IE) systems. IE systems scan articles and retrieve specific information which is required for a certain domain defined in advance. Currently, many IE tasks are performed by pattern matching. For example, if the system receives a sentence "Two more people have died in Hong Kong from SARS," and the system has a pattern "*NUMBER* people die in *LOCATION*" in its inventory, then the system can apply the pattern to the sentence and fill the slots, and obtain information such as "NUMBER = two more, LOCATION = Hong Kong". In most IE systems, the performance of the system is dependent on these well-designed patterns.

In natural language sentences, a single event can be expressed in many different ways. So we need to prepare patterns for various kinds of expressions used in articles. We are interested in clustering IE patterns which capture the same information. For example, a pattern such as "*LOCATION* reports *NUMBER* deaths" can be used for the same purpose as the previous one, since this pattern could also capture the casualties occurring in a certain location. Prior work to relate two IE patterns was reported by (Shinyama et al., 2002). However, in this attempt only limited forms of expressions could be obtained. Furthermore, the obtained paraphrases were limited to existing IE patterns only. We are interested in collecting various kinds of clues, including similar IE patterns themselves, to connect two patterns. In this paper, we tried to obtain more varied paraphrases. Although our current method is intended for use in Information Extraction, we think the same approach can be applied to obtain paraphrases for other purposes, such as machine translation or text summarization.

There have been several attempts to obtain paraphrases. (Barzilay and McKeown, 2001) applied text alignment to parallel translations of a single text and used a part-of-speech tagger to obtain paraphrases. (Lin and Pantel, 2001) used mutual information of word distribution to calculate the similarity of expressions. (Pang et al., 2003) also used text alignment and obtained a finite state automaton which generates paraphrases. (Ravichandran and Hovy, 2002) used pairs of questions and answers to obtain varied patterns which give the same answer. Our approach is different from these works in that we used comparable news articles as a source of paraphrases and used Named Entity tagging and dependency analysis to extract corresponding expressions.

2 Overall Procedure of Paraphrase Acquisition

Our main goal is to obtain pattern clusters for IE, which consist of sets of equivalent patterns capturing the same information. So we tried to discover paraphrases contained in Japanese news articles for a specific domain. Our basic idea is to search news articles from the same day. We focused on the fact that various newspapers describe a single event in different ways. So if we can discover an event which is reported in more then one newspaper, we can hope these articles can be used as the source of paraphrases. For example, the following articles appeared in "Health" sections in different newspapers on Apr. 11:

1. "The government has announced that two more people have died in Hong Kong after contracting the SARS virus and 61 new cases of the illness have been detected." *(Reuters, Apr. 11)*

2. "Hong Kong reported two more deaths and 61 fresh cases of SARS Friday as governments across the world took tough steps to stop the killer virus at their borders." *(Channel News Asia, Apr. 11)*

In these articles, we can find several corresponding parts, such as "*NUMBER* people have died in *LOCATION*" and "*LOCATION* reported *NUMBER* deaths". Although their syntactic structures are different, they still convey the same single fact. Here it is worth noting that even if a different expression is used, some noun phrases such as "Hong Kong" or "two more" are preserved across the two articles. We found that these words shared by the two sentences provide firm anchors for two different expressions. In particular, Named Entities (NEs) such as names, locations, dates or numbers can be the firmest anchors since they are indispensable to report an event and difficult to paraphrase.

We tried to obtain paraphrases by using this property. First we collect a set of comparable articles which reports the same event, and pull appropriate portions out of the sentences which share the same anchors. If we carefully choose appropriate portions of the sentences, the extracted expressions will convey the same information; i.e. they are paraphrases. After corresponding portions are obtained, we generalize the expressions to templates of paraphrases which can be used in future.

Our method is divided into four steps:

1. Find comparable sentences which report the same event from different newspapers.

2. Identify anchors in the comparable sentences.

3. Extract corresponding portions from the sentences.

4. Generalize the obtained expressions to paraphrase templates.

Figure 1 shows the overall procedure. In the remainder of this section, we describe each step in turn.

2.1 Find Comparable Sentences

To find comparable articles and sentences, we used methods developed for Topic Detection and Tracking (Wayne, 1998). The actual process is divided into two parts: article level matching and sentence level matching. Currently we assume that a pair of paraphrases can be found in a single sentence of each article and corresponding expressions don't range across two or more sentences. Article level matching is first required to narrow the search space and reduce erroneous matching of anchors.

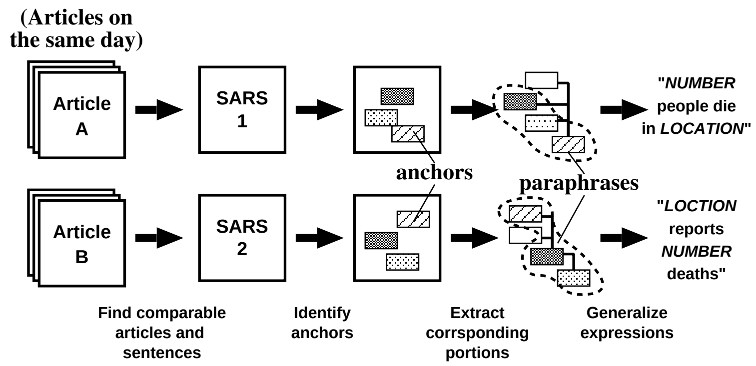

Figure 1: The overall procedure

Before applying this technique, we first preprocessed the articles by stripping off the strings which are not considered as sentences. Then we used a part-of-speech tagger to obtain segmented words. In the actual matching process we used a method described in (Papka et al., 1999) to find a set of comparable articles. Then we use a simple vector space model for sentence matching.

2.2 Identify Anchors

Before extracting paraphrases, we find anchors in comparable sentences. We used Extended Named Entity tagging to identify anchors. A Named Entity tagger identifies proper expressions such as names, locations and dates in sentences. In addition to these expressions, an Extended Named Entity tagger identifies some common nouns such as disease names or numbers, that are also unlikely to change (Sekine et al., 2002). For each corresponding pair of sentences, we apply the tagger and identify the same noun phrases which appear in both sentences as anchors.

2.3 Extract Corresponding Sentence Portions

Now we identify appropriate boundaries of expressions which share the anchors identified in the previous stage. To avoid extracting non-grammatical expressions, we operate on syntactically structured text rather than sequences of words. Dependency analysis is suitable for this purpose, since using dependency trees we can reconstruct grammatically correct expressions from a spanning subtree whose root is a predicate. Dependency analysis also allows us to extract expressions which are subtrees but do not correspond to a single contiguous sequence of words.

We applied a dependency analyzer to a pair of corresponding sentences and obtained tree structures for each sentence. Each node of the tree is either a predicate such as a verb or an adjective, or an argument such as a noun or a pronoun. Each predicate can take one or more arguments. We generated all possible combinations of subtrees from each dependency tree, and compared the anchors which are included in both subtrees. After a pair of corresponding subtrees which share the anchors is found, the subtree pair can be recognized as paraphrases. In actual experiments, we put some restrictions on these subtrees, which will be discussed later. This way we can obtain grammatically well-formed portions of sentences (Figure 2).

2.4 Generalize Expressions

After corresponding portions are obtained, we generalize the expressions to form usable templates of paraphrases. Actually this is already done by Extended Named Entity tagging. An Extended Named Entity tagger classifies proper expressions into several categories. This is similar to a part-of-speech tagger as it classifies words into several part-of-speech categories. For example, "Hong Kong" is tagged as a location name, and "two more" as a number. So an expression such as "*two more* people die in *Hong Kong*" is finally converted into the form "*NUMBER* people die in *LOCATION*" where *NUMBER* and *LOCATION* are slots to fill in. This way we obtain expressions which can be used as IE patterns.

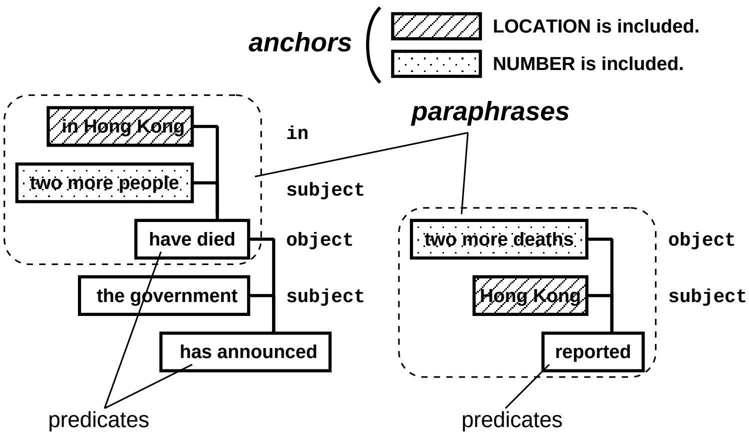

Figure 2: Extracting portions of sentences

3 Handling Problems in Real Texts

In the previous section we described our method for obtaining paraphrases in principle. However there are several issues in actual texts which pose difficulties for our method.

The first one is in finding anchors which refer to the same entity. In actual articles, names are sometime referred to in a slightly different form. For example, “President Bush” can also be referred to as “Mr. Bush”. Additionally, sometime it is referred to by a pronoun, such as “he”. Since our method relies on the fact that those anchors are preserved across articles, anchors which appear in these varied forms may reduce the actual number of obtained paraphrases.

To handle this problem, we extended the notion of anchors to include not just Extended Named Entities, but also pronouns and common nouns such as “the president”. We used a simple coreference resolver after Extended Named Entity tagging. Currently this is done by simply assigning the most recent antecedent to pronouns and finding a longest common subsequence (LCS) between two noun groups. Since it is possible to form a compound noun such as “President-Bush” in Japanese, we computed LCS for each character in the two noun groups. We used the following condition to decide whether two noun groups s_1 and s_2 are coreferential:

- if $2 \leq min(|s_1|, |s_2|) \leq |LCS(s_1, s_2)|$, then s_1 and s_2 are considered coreferential.

Here $|s|$ denotes the length of noun group s and $LCS(s_1, s_2)$ is the LCS of two noun groups s_1 and s_2.

The second problem is to extract appropriate portions as paraphrase expressions. Since we use a tree structure to represent the expressions, finding common subtrees may take an exponential number of steps. For example, if a dependency tree in one article has one single predicate which has n arguments, the number of possible subtrees which can be obtained from the tree is 2^n. So the matching process between arbitrary combinations of subtrees may grow exponentially with the length of the sentences. Even worse, it can generate many combinations of sentence portions which don't make sense as paraphrases. For example, from the expression “two more people have died in Hong Kong” and “Hong Kong reported two more deaths”, we could extract expressions “in Hong Kong” and “Hong Kong reported”. Although both of them share one anchor, this is not a correct paraphrase. To avoid this sort of error, we need to put some additional restrictions on the expressions.

(Shinyama et al., 2002) used the frequency of expressions to filter these incorrect pairs of expressions. First the system obtained a set of IE patterns from corpora (Sudo and Sekine, 2001), and then calculated the score for each candidate paraphrase by counting how many times that expression appears as an IE pattern in the whole corpus. However, with this method, obtainable expressions are limited to existing IE patterns only. Since we wanted to obtain a broader range of expressions not limited to IE patterns themselves, we tried to use other restrictions which can be acquired independently of the IE system.

We partly solve this problem by calculating the plausibility of each tree structure. In Japanese sentences, the case of each argument which modifies a predicate is represented by a case marker (postposition or *joshi*) which follows a noun phrase, just like prepositions in English but in the opposite order. These arguments include subjects and objects that are elucidated syntactically in English sentences. We collected frequent cases occurring with a specific predicate in advance. We applied this restriction when generating subtrees from a dependency tree by calculating a score for each predicate as follows:

Let an instance of predicate p have cases $C = \{c_1, c_2, ..., c_n\}$ and a function $N_p(I)$ be the number of instances of p in the corpus whose cases are $I = \{c_1, c_2, ..., c_m\}$. We compute the score $S_p(C)$ of the instance:

$$S_p(C) = \frac{\sum_{I \subset C} N_p(I)}{\text{the number of instances of } p \text{ in the corpus}}.$$

Using this metric, a predicate which doesn't have cases that it should usually have is given a lower score. A subtree which includes a predicate whose score is less than a certain threshold is filtered out. This way we can filter out expressions such as "Hong Kong reported" in Japanese since it would lack an object case which normally the verb "report" should have. Moreover, this greatly reduces the number of possible combinations of subtrees.

4 Experiments

We used Japanese news articles for this experiment. First we collected articles for a specific domain from two different newspapers (*Mainichi* and *Nikkei*). Then we used a Japanese part-of-speech tagger (Kurohashi and Nagao, 1998) and Extended Named Entity tagger to process documents, and put them into a Topic Detection and Tracking system. In this experiment, we used a modified version of a Japanese Extended Named Entity tagger (Uchimoto et al., 2000). This tagger tags person names, organization names, locations, dates, times and numbers.

Article pairs:

	Obtained	Correct
System	195	156 (80%)

Sentence pairs:
(from top 20 article pairs)

	Obtained	Correct
Manual	93	93
W/o coref.	55	41 (75%)
W coref.	75	52 (69%)

Paraphrase pairs:

	Obtained	Correct
W/o coref. or restriction	106	25 (24%)
W/o coref., w restriction	32	18 (56%)
W coref. and restriction	37	23 (62%)
Manual (in 5 hours)	(100)	(100)

Table 1: Results in the murder cases domain

Sample 1:
- *PERSON1* killed *PERSON2*.
- *PERSON1* let *PERSON2* die from loss of blood.

Sample 2:
- *PERSON1* shadowed *PERSON2*.
- *PERSON1* kept his eyes on *PERSON2*.

Figure 3: Sample correct paraphrases obtained (translated from Japanese)

Sample 3:
- *PERSON1* fled to *LOCATION*.
- *PERSON1* fled and lay in ambush to *LOCATION*.

Sample 4:
- *PERSON1* cohabited with *PERSON2*.
- *PERSON1* murdered in the room for cohabitation with *PERSON2*.

Figure 4: Sample incorrect paraphrases obtained (translated from Japanese)

Next we applied a simple vector space method to obtain pairs of sentences which report the same event. After that, we used a simple coreference resolver to identify anchors. Finally we used a dependency analyzer (Kurohashi, 1998) to extract portions of sentences which share at least one anchor.

In this experiment, we used a set of articles which reports murder cases. The results are shown in Table 1. First, with Topic Detection and Tracking, there were 156 correct pairs of articles out of 193 pairs obtained. To simplify the evaluation process, we actually obtained paraphrases from the top 20 pairs of articles which had the highest similarities. Obtained paraphrases were reviewed manually. We used the following criteria for judging the correctness of paraphrases:

1. They has to be describing the same event.
2. They should capture the same information if we use them in an actual IE application.

We tried several conditions to extract paraphrases. First we tried to extract paraphrases using neither coreference resolution nor case restriction. Then we applied only the case restriction with the threshold $0.3 < S_p(C)$, and observed the precision went up from 24% to 56%. Furthermore, we added a simple coreference resolution and the precision rose to 62%. We got 23 correct paraphrases. We found that several interesting paraphrases are obtained. Some examples are shown in Figure 3 (correct paraphrases) and Figure 4 (incorrect paraphrases).

It is hard to say how many paraphrases can be ultimately obtained from these articles. However, it is worth noting that after spending about 5 hours for this corpus we obtained 100 paraphrases manually.

5 Discussion

Some paraphrases were incorrectly obtained. There were two major causes. The first one was dependency analysis errors. Since our method recognizes boundaries of expressions using dependency trees, if some predicates in a tree take extra arguments, this may result in including extraneous portions of the sentence in the paraphrase. For example, the predicate "lay in ambush" in **Sample 3** should have taken a different noun as its subject. If so, the predicate doesn't share the anchors any more and could be eliminated.

The second cause was the lack of recognizing contexts. In **Sample 4**, we observed that even if two expressions share multiple anchors, an obtained pair can be still incorrect. We hope that this kind of error can be reduced by considering the contexts around expressions more extensively.

6 Future Work

We hope to apply our approach further to obtain more varied paraphrases. After a certain number of paraphrases are obtained, we can use the obtained paraphrases as anchors to obtain additional paraphrases. For example, if we know "*A* dismantle *B*" and "*A* destroy *B*" are paraphrases, we could apply them to "U.N. reported *Iraq* dismantling *more missiles*" and "U.N. official says *Iraq* destroyed *more Al-Samoud 2 missiles*", and obtain another pair of paraphrases "*X* reports *Y*" and "*X* says *Y*".

This approach can be extended in the other direction. Some entities can be referred to by completely different names in certain situations, such as "North Korea" and "Pyongyang". We are also planning to identify these varied external forms of a single entity by applying previously obtained paraphrases. For example, if we know "*A* restarted *B*" and "*A* reactivated *B*" as paraphrases, we could apply them to "*North Korea* restarted *its nuclear facility*" and "*Pyongyang* has reactivated *the atomic facility*". This way we know "North Korea" and "Pyongyang" can refer to the same entity in a certain context.

In addition, we are planning to give some credibility score to anchors for improving accuracy. We found that some anchors are less reliable than others even if they are considered as proper expressions. For example, in most U.S. newspapers the word "U.S." is used in much wider contexts than word such as "Thailand" although both of them are country names. So we want to give less credit to these widely used names.

We noticed that there are several issues in generalizing paraphrases. Currently we simply label every Named Entity as a slot. However expressions such as "the governor of *LOCATION*" can take only a certain kind of locations. Also some paraphrases might require a narrower context than others and are not truly interchangeable. For example, "*PERSON* was sworn" can be replaced with "*PERSON* took office", but not vice versa.

7 Conclusions

In this paper, we described a method to obtain paraphrases automatically from corpora. Our key notion is to use comparable articles which report the same event on the same day. Some noun phrases, especially Extended Named Entities such as names, locations and numbers, are preserved across articles even if the event is reported using different expressions. We used these noun phrases as anchors and extracted portions which share these anchors. Then we generalized the obtained expressions as usable paraphrases.

We adopted dependency trees as a format for expressions which preserve syntactic constraints when extracting paraphrases. We generate possible subtrees from dependency trees and find pairs which share the anchors. However, simply generating all subtrees ends up obtaining many inappropriate portions of sentences. We tackled this problem by calculating a score which tells us how plausible extracted candidates are. We confirmed that it contributed to the overall accuracy. This metric was also useful to trimming the search space for matching subtrees. We used a simple coreference resolver to handle some additional anchors such as pronouns.

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.

References

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the ACL/EACL*.

Sadao Kurohashi and Makoto Nagao, 1998. *Japanese Morphological Analysis System JUMAN*. Kyoto University, version 3.61 edition.

Sadao Kurohashi, 1998. *Kurohashi-Nagao parser*. Kyoto University, version 2.0 b6 edition.

Dekang Lin and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343–360.

Bo Pang, Kevin Knight, and Danial Marcu. 2003. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *NAACL-HLT*.

Ron Papka, James Allen, and Victor Lavrenko. 1999. UMASS Approaches to Detection and Tracking at TDT2. In *DARPA: Broadcast News Workshop*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Satoshi Sekine, Kiyoshi sudo, and Chikashi Nobata. 2002. Extended Named Entity Hierarchy. In *Proceedings of the LREC*.

Yusuke Shinyama, Satoshi Sekine, Kiyoshi Sudo, and Ralph Grishman. 2002. Automatic Paraphrase Acquisition from News Articles. In *Proceedings of the Second International Conference on Human Language Technology Research*.

Kiyoshi Sudo and Satoshi Sekine. 2001. Automatic Pattern Acquisition for Japanese Information Extraction. In *Proceedings of the HLT*.

Kiyotaka Uchimoto, Masaki Murata, Qing Ma, Hiromi Ozaku, and Hitoshi Isahara. 2000. 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.

Charles L. Wayne. 1998. Topic Detection & Tracking: A Case Study in Corpus Creation & Evaluation Methodologies. In *Proceedings of the LREC*.