TowardsAutomaticAcquisitionofPatternsfor
Information Extraction

Chikashi NOBATA
Department of Information Science, University of Tokyo
Science Building 7, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, JAPAN
E-mail: nova@is.s.u-tokyo.ac.jp

Satoshi SEKINE
Computer Science Department, New York University
715 Broadway, Room 709, New York, NY 10003, USA
E-mail: sekine@cs.nyu.edu

Abstract
One of the hardest obstacles in developing Information Extraction (IE) systems is the
portability of the systems. Many IE systems use a pattern-matching approach, but the
patterns have to be created for each target task or target domain. In order to improve
IE systems and make them portable to various domains, we should have a mechanism
to create pattern easily. We will propose a method to acquire such patterns almost
automatically from a large untagged corpus. First, the system collects sentences which
match the target task and the domain. Next, it adds the information to them, such
as part-of-speeches and tags for proper names, dates, currency. Then, the sentences
are clustered and generalized based on their similarity to become useful patterns
for IE. We applied the method to Japanese newspaper articles about the succession
of company executives, and compared the performance based the acquired patterns
with that based on manually created patterns. We will describe the limitations and
potential of the present method, and suggest future ways of improving it.

1 Introduction
There has been a growing interest in developing systems for Information Extraction (IE)\(^1\). The objective
of IE systems assumed in this paper is to identify instances of a particular class of events or relationships
in a natural language text, and to extract the relevant arguments of the event or of the relationship. The IE
task therefore involves the creation of a structured representation (such as a database) of selected informa-
tion drawn from a text. So, the task of IE is both simpler and more tractable than the more ambitious
problem of general text understanding. Furthermore, it involves a reasonable evaluation method of performance
and progress in the field. Indeed, over the last decade, competing IE systems have undergone a series of formal evaluations — especially in the Message Understanding Conferences (MUCs)\[^3\].

The technique used in most of the systems is to prepare a set of patterns for the specific task. A new
input text is examined against these patterns and information is extracted based on the matched pattern.
It works well, but there is a serious bottleneck in this method. We have to create patterns and, furthermore, some patterns are likely to be scenario dependent. So, we have to develop a set of patterns for each scenario. Although some systems provide a good user interface or semi-automatic methods to facilitate the task, the preparation is done mainly by hand and it is tedious. Some of these systems will be mentioned later.

In this paper, we will propose an almost automatic method to acquire patterns useful for IE from a large
untagged corpus. The basic idea is the following:

1. From a large corpus, collect sentences which are related to the scenario and likely to be a source
   of patterns.

2. Run low level processes, i.e. annotate sentences with part-of-speeches(POS) and Named Entity\(^2\) tags.

3. Cluster the sentences based on a similarity mea-

\(^1\)In this paper, the definition of IE is narrower than some; for a review of the fuller range of systems falling under this
category, the reader is referred to [9].

\(^2\)In this paper, the definition of IE is narrower than some; for a review of the fuller range of systems falling under this
category, the reader is referred to [9].
sure and generalize sequences of tokens based on their similarity.

Obviously, this method has limitations. For example, the current strategy cannot handle information expressed across multiple sentences. We have to have taggers which annotate Named Entities, and it requires some human intervention. However, this method has great potential, because, as it is basically an unsupervised method, we have virtually no limitation on the size of the training corpus, and hence we may be able to reduce the coverage problem greatly.

2 Information Extraction

When referring to IE in this paper, we will use the terms accepted in the MUC literature. “Domain” will denote the class of textual documents to be processed, such as “business news,” while “scenario” will refer to the set of facts to be extracted, i.e. the specific extraction task that is applied to documents within the domain. One example of a scenario is “executive succession”, the task tested in MUC-6, where the system seeks to identify events in which corporate managers left their positions or assumed new ones. Thus, the events sought in this task would be persons, companies, and geographic locations; an example of a relationship of interest is the relationship of employment between a person and a company, or the relationship of location between a company and a city; an example of an event is the hiring of a person, along with, for example, the post assumed, the reason for the vacancy, and the company from which the person came. Figure 1 shows a (slightly modified) text segment from the MUC-6 development corpus, with the corresponding table of extracted information. Each row in the table corresponds to an event, and the columns correspond to the slots in the event.

In order to extract the information from texts, we have to have patterns representing entities and events in the texts. Figure 2 shows some of the patterns, manually created for extracting information from Japanese articles. The objective of the method described in this paper is to acquire such patterns as listed in Figure 2.

3 Related Work

There has been a good deal of work which aims at easing the burden of pattern creation.

---

\*Named Entity task is defined by MUCs. The task is to insert SGML tags into the text to mark each string that represents a person, organization, or location name, or a date or time stamp, or a currency or percentage figure. In this paper, a position in an organization is also marked.

4 Algorithm

4.1 Overview of the Algorithm

Figure 3 shows the overview of the algorithm. The pattern set is created from a large untagged corpus. Keywords are used to collect a subset of the corpus, and again a subset of the data is selected based on the subject line of the article. POS tagging and Named
Entity tagging are performed for each sentence in the corpus. Sentences with important keywords and with specific Named Entities are extracted to be the source of the patterns. Then the tagged sentences are merged and generalized based on their similarity.

![Diagram of the algorithm](image)

**4.2 Description**

In this subsection, the algorithm is described in detail.

**Article Extraction by Keywords and Selection**

First, keywords are considered, based on the given scenario. It can be done by consulting a thesaurus or browsing through articles. The articles of the given scenario are retrieved based on the keywords. These articles include at least one keyword. The subject line of the articles is scanned, and we find that there are some common subjects which articles are not related to the scenario. Actually, in this experiment, two very frequent expressions in subjects found were not suitable for IE from sentences; these expressions imply the articles don’t contain sentences, but only a list of name of people who were promoted. These types of articles were deleted from the data.

**POS and Named Entity Tagging**

POS tagging and Named Entity tagging are performed on the sentences in the selected articles. Since words are not separated in Japanese, the morphological analysis is also performed at this stage. We used JUMAN[5], their POS information is used in the Named Entity tagging process. The Named Entity tagging marks up Named Entities (like organization, person, location, position names), which are important in the business news domain. We used a system which employs a decision tree algorithm([7], [8]). At this point, sentences which are likely to be related to the scenario are retrieved. Only the sentences with frequencies more than one and which contain one of the keywords and specific named entities are used in the next process.

**Merging Patterns**

This is the main part of our system. We make a new pattern by merging the most similar pair of patterns, iteratively. At the initial stage, each tagged sentence is regarded as a pattern consisting only of lexical items and named entities. The merging is done by combining the most similar pair of patterns into one pattern. There are various methods to define similarity, but for the moment, we are using a simple method. We take the number of different items in two patterns divided by the total number of items in the two original patterns. The smaller the value, the more similar the two patterns. When merging two patterns, we may have to do two kinds of operations, unless these two patterns are exactly the same. (This does not happen at the initial stage, but it could happen in a later stage.) The two operations are “exchange” and “ignore”. “Exchange” is used when two corresponding items are the only difference between the two patterns. “Ignore” is used when one of the patterns has extra items. Figure 4 shows simple examples of merging operation. (the first one is an example of “exchange”, the second one is an example of “ignore”). When we create a cluster...
\[(A \ B \ C), (A \ D \ C) \rightarrow A \text{ exch } C \quad (\text{exch} = B \ | \ D)\]
\[(A \ B \ C), (A \ C) \rightarrow A \text{ ignr } C \quad (\text{ignr} = B)\]

Figure 4: Merging Operation

by an “exchange” operation, all the occurrences of the clustered items in the pattern set are replaced by the cluster. The merging iteration starts from the set of patterns with all tagged sentences considered as individual patterns. After each merging process finishes, two old patterns are deleted and one new pattern is created. So the merging decreases the number of total patterns by 1, and the set of patterns is gradually generalized. The merging process can be repeated until the number of patterns becomes 1. In the entire process, we will have as many pattern sets as tagged sentences, from the set of unmerged sentences to one complex pattern. We will need to find a criterion to choose the suitable set among them in the future, but currently, we evaluate the performance of all the sets in the following experiment.

5 Experiment

5.1 Task

The task in this experiment is to find out information about an executive succession event, which is the same as the scenario used in MUC-6. The experiment is conducted for Japanese articles and Nikkei-94 is used. Eighteen keywords are used in the experiment, as shown in Figure 5.

Figure 5: List of Keywords Used for the Succession Scenario

5.2 Data

We used Nikkei-94 data distributed by the LDC for the experiment. The data size (number of articles) in each process is listed in Table 1. We did not use all the selected articles for the training. The remaining articles are used for the evaluation, the manual referencing described in a later section, and the future development.

The data size related to sentences and patterns is listed in Table 2. We retrieved 13,554 sentences from 10,612 training articles. Among them, there are 114 types of sentences which are frequency more than one and 19,182 sentences with frequency one. We used only the 114 sentences for the automatic pattern creation to avoid noises. The total frequency of these sentences is 446. The best result (described later) was achieved when the number of patterns ranges from 52 to 79.

<table>
<thead>
<tr>
<th>Description</th>
<th>Num. of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nikkei-94</td>
<td>192,969</td>
</tr>
<tr>
<td>Retrieved articles</td>
<td>15,910</td>
</tr>
<tr>
<td>Selected articles</td>
<td>11,985</td>
</tr>
<tr>
<td>Training articles</td>
<td>10,612</td>
</tr>
<tr>
<td>Evaluation articles</td>
<td>203</td>
</tr>
</tbody>
</table>

Table 1: Data Size (article)

<table>
<thead>
<tr>
<th>Description</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved sent.</td>
<td>13,554 (instance)</td>
</tr>
<tr>
<td>Tagged sent. (freq\geq1)</td>
<td>13,182 (type)</td>
</tr>
<tr>
<td>Tagged sent. (freq\geq2)</td>
<td>114 (type)</td>
</tr>
<tr>
<td>The Best Pattern</td>
<td>52-79 (type)</td>
</tr>
</tbody>
</table>

Table 2: Data Size (sentence and pattern)

5.3 Example

Figure 6 shows an example of the merging of patterns. This merging occurred at the 63rd iteration.

The new pattern (A) is created from two patterns (B and C) which have already been partially generalized. Pattern A is created by merging Pattern B and C. Both pattern B and C describe a retirement event, so does pattern A. The third word in Pattern A (は) means “resign”. “C-#” represents a cluster of elements (words). For example, C-1 is created by merging COMPANY and C-3. Both of them include a COMPANY item, although semantically these expressions are slightly different. C-1 is a cluster which already contains both “は” and “が”, all of the words in the cluster are semantically similar. The “?” after the C-2 in Pattern A indicates that it is optional. The words in C-2 are also similar, although the first word is semantically different from the others.

This is an interesting example, and indeed intuitively appropriate.

5.4 Manually Created Patterns

For the comparison, we created a pattern set by hand. This is done by referencing a small corpus (87 articles), which is notably smaller than the training data from the experiment (10,612 articles). It took about 0.2 person month effort, including the creation of the logical form. The manual patterns contain low level patterns (like noun groupings), and high level
Merged Patterns:

A: C-1 PERSON POSITION C-1 退任 C-2?。
B: COMPANY, PERSON POSITION が 退任。
C: C-3 PERSON POSITION は 退任 C-2。

Sub Patterns:

C-1: C-3 [COMPANY,]
C-2: した (past tense indicator) |
する (present tense indicator) |
する見込み (“It is planned to ...”) |
する予定 (“It is scheduled to ...”) |
C-3: COMPANY出身の (a person who previously worked at COMPANY)
C-4: は [subject indicator] |
が [subject indicator] |
、(comma) |
の [subject indicator when modifying noun (as one usage); similar to “of”; person の retirement = retirement of person]

Figure 6: Example of Merging Pattern

patterns (like clause level patterns). These are applied successively from the lower level to the higher level. This strategy is clearly different from the strategy taken in our experiment, in which only sentence patterns are used. The number of manually created patterns is 91, and simple examples of them are shown in Figure 2.

5.5 Result

Table 3 shows the result of the experiments. The result was calculated using the total number of successive events detectable from matched patterns. Even several sentences are matched by patterns, we only count one if these sentences express the same event. This metric is motivated by the MUC evaluation method, which is based on the number of information about the events. However, we cannot use such a measure, and we just counted up the number of events possibly extracted from the text. The result shown for the automatically extracted patterns is the best result using the patterns. It’s about 5% lower than the result of the manual pattern, but it took a long time to create manual patterns, we believe the result is reasonable.

Table 3: Result of Experiments

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Automatic</td>
</tr>
<tr>
<td>Recall</td>
<td>31.67</td>
</tr>
<tr>
<td>Precision</td>
<td>75.08</td>
</tr>
<tr>
<td>F-measure</td>
<td>44.19</td>
</tr>
</tbody>
</table>

Figure 7 shows the relationship between recall and precision using the different pattern sets. The line indicates the relationship between recall and precision of the automatic patterns varying the number of patterns. The circle(○) indicates the values the pattern sets have actually showed. In fact, 114 sets of patterns are divided into 6 classes, and all the sets in one class have the same performance in matching. The dot(●) indicates the performance of the manual patterns. Patterns can match not only the whole sentence, but a part of a sentence. So, the tagged sentences, which are considered as a set of unmerged patterns, show higher recall than their frequency in the training data. As patterns are being generalized in the merging process, their recall is increasing. However, excessive generalization makes their precision decrease.

6 Discussion

The work described in this paper is merely a first step in the exploration of this method. First of all, we would like to apply our method to another scenario to explore its general applicability. Also, we are planning to expand it in several directions, which include:

• Article/sentence extraction technique
  We retrieved articles from a corpus using key-
words and selected relevant articles by looking at the subject line. Obviously, this is a heuristic method and it should be more automatic. Also, in the extracted sentences, we found a lot of noise. For example, some sentences were extracted that contained no successor event. The means to extract only relevant articles and sentences needs to be discovered.

- **More generalization at a lower level**
  For example, POS information or pre-defined word clusters can be used in the patterns. We are planning to try automatic generation of word clusters based on the corpus of the corresponding domain or scenario, because clusters might vary across different domains or scenarios. Also, compound words, in particular verb suffix compounds, should be introduced which will make it easier to generalize.

- **Utilization of existing knowledge**
  We can utilize existing knowledge, like dictionaries and thesauri. This usage might introduce domain dependency problems and sense ambiguity. It is worthwhile investigating.

- **Definition of similarity**
  In the experiment, the similarity measure was very simple. However, we believe that we can find a similarity definition which is more suitable to this purpose.

- **Single frequency sentences**
  In the experiment, we did not use sentences with frequency one (there are more than ten thousand). It is certain that these sentences contain noises, but they contain useful information as well. We would like to find a method to utilize this information. Also, a method which can handle such a large set has to be developed.

- **Association with a logical form**
  The described method only extracts patterns which are useful for IE tasks. The ultimate target is to create patterns which extract event information. Connection to a logical form is needed. This touches on the issue of semantic differences of clustered words shown in Figure 6. We may utilize an existing thesaurus.

**Acknowledgments**

We would like to thank our colleagues at New York University and University of Tokyo. In particular, Prof. Ralph Grishman and Roman Yangarber gave us help and comments, which are substantial parts of the work. We also thank Catherine Macleod for her support.

**References**


