Knowledge Discovery using Robust Natural Language Processing

Tohru Nagano, Koichi Takeda, and Tetsuya Nasukawa
IBM Research, Tokyo Research Laboratory, IBM Japan
1623-14 Shimotsuruma, Yamato, Kanagawa, Japan, 242-0007
TEL:+81-46-215-4438 FAX:+81-46-273-7428
{tohru3, takedasu, nasukawa}@jp.ibm.com

Abstract

Large text databases potentially contain a great wealth of knowledge. However, text represents factual information (and information about the authors' communicative intentions), in a complex, rich, and opaque manner. Consequently, unlike numerical and fixed field data, it cannot be analyzed by standard statistical data mining methods. Relying on human analysis results in either huge workloads or the analysis of only a tiny fraction of the database.

We are developing a method of robust natural language processing for text mining that aims to find regularities/associations in textual data by focusing on an ontology of significant terms and dependencies to capture the textual content. We will show that text mining can also be used to facilitate information retrieval by providing aggregate information as a content-based overview of the underlying textual databases.

keyword: Text Mining, Shallow Parsing Knowledge Discovery

1 Introduction

It is well known that a wide range of knowledge can be extracted from textual data, such as linguistic knowledge for NLP (Natural Language Processing) (Knight, 1999) and that domain-specific lexical and semantic information may be stored in a database (Hahn and Schnattinger, 1997). Such linguistic and conceptual entities are used to capture trends and frequently asked questions in the text.

As shown in (Mladenic, 1999), most of the research to date on handling textual content in large textual databases has employed bag-of-words techniques. However, it is hard for a user to capture textual content only from a discrete set of keywords. Labeling a cluster with just nouns and noun phrases is sometimes misleading because a problematic sentence (*"modem is broken"*) and its negation (*"modem is not broken"*) are easily mixed in such a cluster. In other words, the bag-of-word approach is not sufficient for knowledge discovery.

Text mining is a technology, analogous to data mining (Agrawal et al., 1993), for knowledge discovery and for finding hidden regularities and associations from a large collection of textual data. In data mining research, finding association rules are well known technology. Therefore, we tried this technique to extract knowledge with a bag-of-words approach. The following associations shown in Table 1 are unusual correlations (↔ in the table indicates the correlations between keywords), resulting from mining mining an actual real customer call center's data using noun groups. (Original is in Japanese.)

Table 1: Association Rules using Noun

Keyword1		Keyword2
English version	↔	English
voice	↔	type
application	↔	CD
Hard disk	↔	ROM
TP365XD	↔	365XD

Most of the correlations show the elements of compound nouns. Some correlations, such as the term *"voice"* and the term *"type"* in Table 1 should be a compound noun *"voice type"* (a product's name) . These are useful as linguistic knowledge, but, this kind of linguistic knowledge is not what we want. Significant terms and dependencies, however, can successfully identify the content (i.e., *modem-broken* and *modem-not-broken*). The Table 2 shows relations (⇒ means the dependency between words) using our methodology. This result that captures the context is more informative than the above result.

Table 2: Dependencies using Verb and Noun

Keyword1		Keyword2
`safe mode`	⇒	`possible to start up`
`display driver`	⇒	`not found`
`Backup CD`	⇒	`on sale?`
`screen`	⇒	`freeze`
`memory`	⇒	`add`

In this paper, we propose a schema for a text mining system without a large amount of background knowledge, and describe a notion of significant terms and dependencies (noun-noun and noun-verb relationships) as informative entities suitable to represent the contents of textual data. We show that from a collection of 43,000 customer claims (one month's data), our significant terms and dependencies can improve the content aggregations better than when using keywords.

2 Information Extraction for Text Mining

2.1 Problem

The terms "text mining" or "text data mining" (Hearst, 1999) have been widely used as they may represent information retrieval systems, which detect and extract proper documents the we want from vast amount of documents, or they may refer to clustering or classification systems, which may break up and organize vast amount of documents. However, most of the state-of-the-art techniques which are called text mining, are not concerned about a lot of information that is conveyed with functional word and relationships between words, or requires a lot of back-ground knowledge such as a large amount of training corpus.

We used customer call center's data in this paper. We want to extract useful knowledge for detecting problems and analyzing customer's requests from such large collection of text data.

Therefore, we are developing methods for extracting information that:

- generate informative terms,
- are robust in the presence of ill-formed textual data,
- are fast even for a large amount of textual data, and
- do not require large lots of back ground knowledge.

2.2 Related Work

In order to represent the content of a document as a whole, some terms or some structures have to be extracted from the text using NLP.

Bag-of-Words

The simplest method of extracting information is by extracting the simple terms, a set of keywords in the document, which is a called bag-of-words approach. When a text *"I want to install IBM network interface card, but I don't know the way."* is given, these 15 words (8 content words) are extracted using a POS tagger (Figure 1).

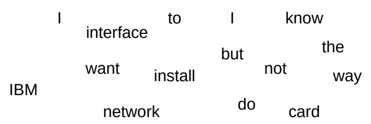

Figure 1: Bag-of-Word

A vector space model (Salton, 1983), which treats a document set as bag-of-words and represents the content by the distribution of the words, has been used for analyzing the similarity of a vector space model.

The advantages of the bag-of-words approach are that it is easy to extract information, and it is robust even when dealing with sparse data. Therefore, generic search system use this approach. However, most of the information for understanding the context of the given text is lost, since the order of the words sequence and the modify-modifiee relations are ignored in this approach.

Word Sequence

(Fujino and Arimura, 2000) use the sequence of words as the representation of a document to find frequent word association patterns in the documents.

However, according (Matsuzawa and Fukuda, 2000), only 40% of the predicate and noun pairs within the same sentence have grammatical dependencies.

Parse Tree

In striking contrast to bag-of-word approach, full parser could generate tree structure (Figure 2).

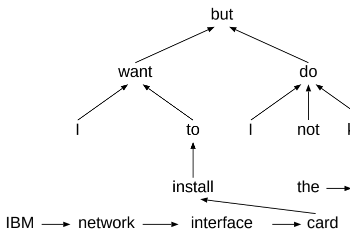

Figure 2: Parse Tree

The advantages of using the parse tree are that it has rich information to represent the content of a document. However, it is hard to statistically analyze.

Patterns

Another method to extract terms from text is a pattern-based approach. This method is used to find specific relationships such as *Question* and *Answer* and the event extraction (Grishman, 1997). However, it is hard to apply this method to text mining. Text mining aims to find hidden regularities and associations from a large collection of textual data. Different from information extraction, for text mining, the type of information which a user requires is not given in advance, the patterns can not be prepared. Moreover, since the sentences from the call center are not written in correct grammar, it is difficult to build the patterns. For example, the grammatical subject is often omitted.

2.3 Basic Idea

The above approaches are not suitable for text mining. Our text mining aims to find hidden regularities and associations in textual data.

The significance and frequency which text mining requires, are alternatives, because text mining is a linguistic and statistical application. In order to gather significant terms for text mining, high frequency terms should be specialized. Low frequency terms should be aggregated with synonyms, which may be found in thesauri. However, the thesauri may not contain low frequency terms since there is usually an enormous number of compound nouns. Figure 3 shows the basic concept used to gather meaningful terms.

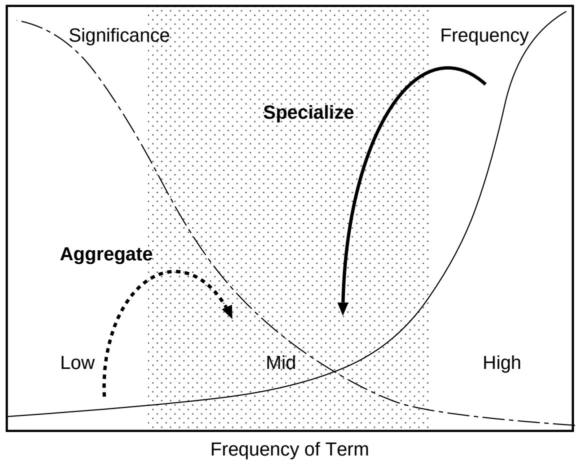

Figure 3: Basic Idea

In order to make use of information at the level of its context and gather informations from the parse tree, we produce simplified semantic parse trees which consist of:

- For noun phrases, to specialize high frequency nouns, we try to make compound nouns from element noun words.
- For verb (adjective) phrases, to specialize verbs, we try to extract the intentions which are contained in modal verbs or functional words.

For example, the syntactic parse tree in Section 2 would be modified to use verb phrases and noun phrases. These phrase boundaries may different from the original syntactic boundaries.

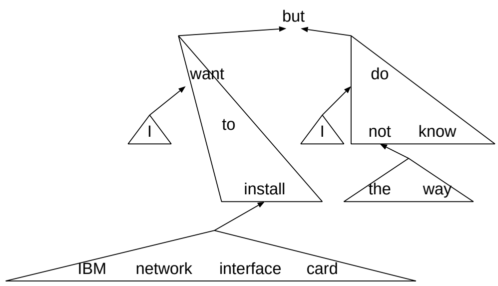

Figure 4: Grouped Parse Tree

As a result of this kind of parsing, intention analysis and term extraction, the parse tree is modified into a tree on this simplified form:

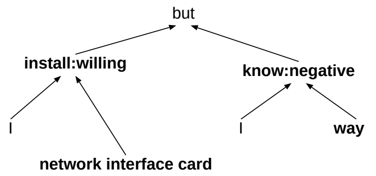

Figure 5: Simplified Parse Tree

This modification and the representation of the sentence make capturing the sentence level context easy. For instance, these three sentences are easily mixed using bag-of-word approach, because all of these sentences contain same content words.

- *X did fail, Y did succeed.*
- *X did not fail, Y did succeed.*
- *Y did not succeed, Did X fail?*

Since such grammatical dependencies may convey useful meaning such as a description of the actions of the subject, the representation of such dependencies are important.

3 Characteristics of Term

In order to investigate what kind of terms are significant, we examined the significance of terms by comparing various sets of terms. The text parts of all of the documents are tokenized and part-of-speech tagged. Then a parser does a dependency analysis. The precision of the parser is not as good as used for other NL applications which require precision, such as machine translation, but it is sufficient for our statistical analysis.

3.1 Customer Call Center Data

We have been focusing on customer call center documents as a first application for text mining. The experiments described in this paper use PC call center's data. Each document (Table 3) is written in Japanese, and includes a Transaction ID (automatically filled in), the Date (automatically filled in), several formatted fields, title of the dialog, and dialogue.

The content of most of these fields is filled in by the call taker. The data in the formatted fields is chosen from selection lists by the call taker. Each selection list contains from 10 to 30 choices. For instance, the field Component contains 30 choices such as Windows95, Memory, Hard Disk, etc. Call takers summarize the dialogues with the customers, and type it into the dialogue field. This text part, which is denoted as Call in the following sample data, is divided into the customer's question and the call taker's response by "*Q:*" and "*A:*". Our study looks only at the questions in the data.

Our test collection contained $43,110$ documents, and each document contained $\simeq 14$ compound nouns. The number of different terms (simple nouns) is $31,609$, and the total number of occurrence of the terms is $1,119,039$. The number of different compound nouns and simple nouns is $600,494$. Therefore a compound noun has $\simeq 2$ component words.

The textual part of call center's data has these feature :

- a large amount of textual data,
- frequent unknown words,
- frequent spelling and typographic errors

Thus, we cannot expect correct semantic parsing with the current state of the art technology that assumes well-formed sentences, and this makes it hard to apply any NLP system based on complete syntactic analysis

3.2 Measure of Significance

As a measure of significance, we use the entropy value of the terms. The entropy value is used to measure the dispersion of the data set into a given range of the data set or the given categories. (Resnik, 1995) used the entropy value to measure the semantic similarity between terms.

All documents have been categorized by the call takers in terms of a Component, a Call-Type and an Answer-Type. The Component field has over 30 category values, and is divided into software/hardware components such as Windows95, Memory, Hard Disk. In the experiments of this section, the category Component is used.

Several methods to measure significance of word have been researched such as *tf/idf* (Salton, 1983) and baseline-method (Hisamitsu et al., 2000). Since we assume that a meaningful term is a term that can be used for determining the category value assigned by the call taker,

Table 3: Sample Record of Call Center's Data

`ID`	`03240210233`
`Date`	`10/21/1999`
`Component`	`Windows95`
`Call-Type`	`Software Problem`
`Answer-Type`	`Information needed`
`Title`	`DVD-ROM`
`Call`	`Q: After upgrading the software, I couldn't play a DVD-ROM. DVD drive is recognized. After inserting DVD-ROM, click the drive icon, then message ''No detected'' appear.` `A: Can you re-install the DVD-ROM drive device driver? First, delete old DVD-ROM driver on the device manager. Then, reboot your computer, and Plug-in-Play detect the new device. Indicate C:/Windows/Driver as the source of the driver.`

we use the entropy value as a measure of significance.

The entropy of terms w is defined as:

$$H(w) = \sum_{i=0}^{N} P(c_i \in w) \log_2 \frac{1}{P(c_i \in w)}$$

$P(w)$ is the probability of w being categorized into category c_i (i.e. Windows95, Memory, Harddisk, and so on). N is the number of categories.

Redundancy is given as:

$$r(w) = 1 - \frac{H(w)}{\log_2 r}$$

where $log_2 r$ gives the upper limit of $H(w)$

This value, $H(w)$, indicates the diversity of term distribution. A lower value means a more diverse term distribution. The value $r(w)$ is the normalized and reversed value of $H(w)$. If this value is high, then the term is a significant term.

Noun, Compound Noun

A noun phrase consists of a sequence of nouns. If a noun phrase has N elements, the number of possible candidates of compound nouns is $N(N+1)/2$. For example, the noun phrase "*Microsoft Internet Explorer*" has three elements, so possible candidates for compound nouns are "*Microsoft*", "*Internet*", "*Explorer*", "*Microsoft Internet*", "*Internet Explorer*" and "*Microsoft Internet Explorer*". We call "*Microsoft Internet Explorer*" a long term, and "*Microsoft*", "*Internet*", "*Explorer*" short terms. If the noun phrase has only one element, the long term and the short term are the same.

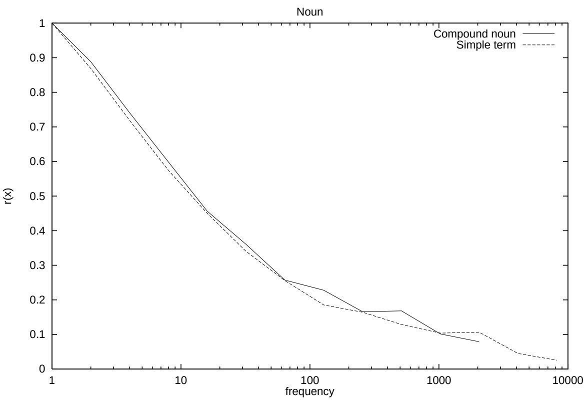

Figure 6: Significance of Noun and Compound Noun

Figure 6 shows the comparison of the frequency of terms (log scale) versus $r(S)$ for long and short terms. A set of terms S contains terms w which have the same frequencies. This value is averaged by the value of the terms which are included in the log-scaled frequency range. The terms with the same frequency have the same significance for both simple terms and compound nouns.

Dependencies

The dependencies between phrases are analyzed by the parser. Figure 7 shows the comparison between the dependencies of noun phrases and noun phrases, and the dependencies of noun phrases and verb phrases. In this case, the noun-verb dependencies are more significant than the noun-noun dependency.

The dependency is quite effective for repre-

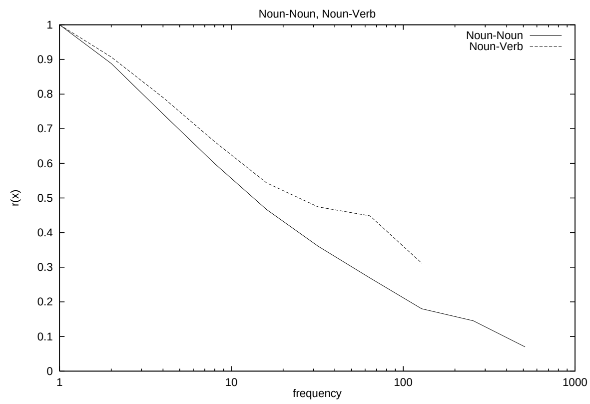

Figure 7: Significance of Dependency

senting sentential level information described in various expressions. For example, in one month data from the Japanese customer call center, 55 cases contained a dependency *"file"* $\Rightarrow$ *"not found"*. Among the 55 cases, only 13 cases contained the same surface expression to describe that context. The dependency representation has been recognized as especially effective for searching for FAQ, due to its capability for representing sentential level context.

Verb

For verb phrases, we investigate the distribution of the modality in verb phrases. We extract over 20 kind of modality (negation, possibility, request, question ...), and then important combinations of all possible combinations are extracted and recognized as intentions. Table 4 shows the distribution of the modality for the verb *"use"* (*"tsukau"* in Japanese) phrase. About a half of the *"use"* verb phrases contain some kind of modality and more than 20% of the verb phrases involve negations of *"use"*.

4 Natural Language Processing

4.1 NLP Overview

Text mining shows an overview of the entire data set. Thus the result of text mining has to be linguistically informative and statistically significant.

The NLP part of system has some functions to extract informative and significant terms from the document set. The system performs dependency analysis, intention extraction, and term extraction. In the case of noun phrases, the process will be done statistically. For verb phrases, intention extraction is performed.

For a part-of-speech tagger, we used JMA (Japanese Morphological Analyzer) (Maruyama and Ogino, 1994). JMA also includes a tokenizer. The accuracy of this parser is more than 99% according to creator's experiments on newspaper articles. This parser also identifies phrase boundaries. The sequence of terms delimited by sentence boundary information is parsed and analyzed by the Intention Extractor

Table 4: Distribution of Modalities

Word	P	N	W	Q		
use					1998	56.2%
can not use	○	○			637	17.9%
can use	○				297	8.4%
want to use			○		262	7.4%
can use ?	○			○	137	3.9%
don't use		○			137	3.9%
use ?				○	57	1.6%
can't use?	○	○		○	19	0.5%
(others)					10	0.3%
Total					3554	100%

```
P   intention ''possibility''
N   intention ''negation''
W   intention ''request''
Q   intention ''question''
```

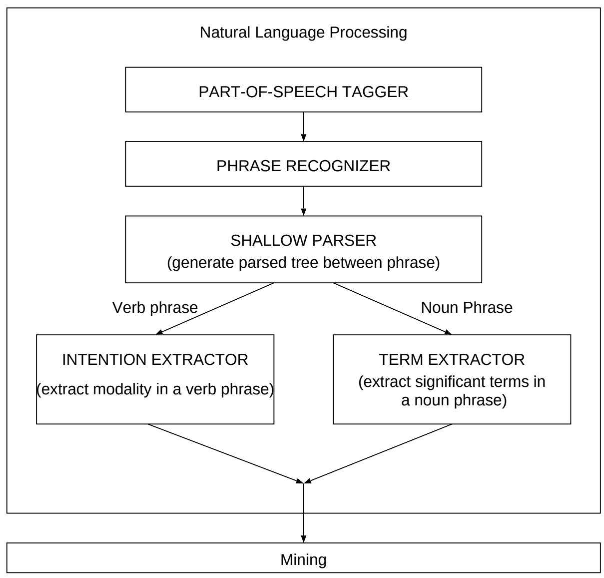

Figure 8: NLP Overview

and the Term Extractor.

4.2 Shallow Parser

Extracting dependencies for keyword expansion (Strzalkowski and Vauthey, 1992) has already researched. They extracted head-modifier pairs using a full parser and a suffix trimmer, in order to measure the relative strength of the connections between the words in syntactic pairs. We extract the dependencies, in order to get more informative relations.

In order to generate the dependencies between phrases, we developed a shallow parser. One of the reasons that we don't use a full parser, but only a shallow parser, is to generate the dependency information quickly, since text mining deals with a large number of sentences. Our test collection contains about 0.21 million sentences in one month's data. Another reason is that the length of the sentences (the number of phrases in a sentence) in a dialogue such as these call center logs is shorter than in formal printed text such as newspaper articles. One sentence contains an average of about nine noun and verb phrases. In addition, the dialogue of the call center is not written with perfect syntax, since the main goal of the call takers is not to write dialogs, but to solve the customers' problems.

4.3 Intention Extractor

When a phrase is a verb phrase, this module is used. This module recognizes the modalities which are included in a verb phrase, and assigns them to various kinds of intention: negation, possibility, request, question and so on.

The verb phrase is represented as a combination of base form of verb $word_{base}$ and the vector of intentions int_i.

$$VP = [word_{base}, (int_0, int_1, int_2, .., int_N)]$$

where int_i is the occurrence of intention i in a phrase, N is the number of the variety of int.

For a simple example, a question mark *"?"* indicates question. The rules to extract modality are described using terms which indicate modality or a correlation of terms. These modalities are extracted by a rule-based pattern matching engine. For example, a sentence *"Can I buy a computer?"* is given. *"Can"* indicates intention possibility and *"?"* indicates question.

The surface of the VP is calculated using $word_{base}$ and int_n. This calculation involve some linguistic rules that can gather some phrases which have some different surface and same intentions. In a case of negation, if the $i_{negation} = 2$ (a double negative), the surface of the VP is not modified. Thus, the verb phrase in this sentence will be assigned compound modality possibility + question, and the representation of it will be modified to *"can buy?"*.

4.4 Term Extractor

The Term Extractor generates domain-specific compound nouns from given noun phrases. When a phrase is a noun phrase, this module is called. This module tries to make simple nouns into as compound nouns. The lists and rules of compound nouns are given as the sequences of terms or as the sequences of part-of-speech.

$$< (word_1|pos_1), (word_2|pos_2), .., (word_N|pos_N) >$$

If there is a rule $< firstname, lastname >$ in given rules, a sequence of term which has the same sequence of part-of-speech : *"John" "Smith"* is recognized *"John Smith"*.

As a result, if a phrase has three words, and $word_1$ and $word_2$ are recognized as a compound word. The noun phrase is represented as the set of two words.

$$NP = \{(word_1 word_2), (word_3)\}$$

In order to create these rules with term level automatically, we use the metrics described below. To extract significant terms:

1. Count the number of element terms w_i, w_j in all the text samples. $(1 < i, j \leq d)$ where d is the number of distinct words.
2. Count the number of bi-gram $(w_i w_j)$ in all the text samples.
3. Extract the frequent bi-gram $(w_i w_j)$ which exceeding the threshold N_{high} as candidates of compound noun.
4. If the re-calculation of the significance of the unified term $r(w_i w_j)$ is larger than the average significance of $r(w_i)$ and $r(w_j)$, it is designated as a compound noun $w_i w_j$.

5. The above steps are repeated for each frequent terms.

In this method, the thresholds must be formulated and validated.

This method can use all terms in the text. The state-of-the-art methods to extract terminology which use term weighting are not suitable for statistical analysis, so they are not directly relevant to our research.

5 Experiment and Evaluation

Clustering is a technology to divide an uncategorized data set into some clusters, and it is useful to provide an overview of a large data set. In order evaluate to effectiveness of our method, we applied our method to cluster labeling.

5.1 Preparation

To compare with the bag-of-word approach, we prepared two different sets from one months' data which contained 43, 110 documents.

1. Noun - set of the noun syntactic phrases.
2. Dependencies - set of the dependencies between noun phrase and verb phrases. These phrases are generated using our methods. A noun phrase is a part of a syntactic noun phrase. A verb phrase is the head of the verb phrase along with some modalities that stand for intentions.

The number of different nouns was 25, 572, and the number of different dependencies was 55, 842.

We use the *single k-means* clustering method to discover clusters. Each feature set for clustering is a set of frequent 500 noun/dependency. Figure 9 shows the number of clusters versus the average distance that is the overall root-mean-squared distance from the record (document) to the centroid. In this figure, the dispersion of the clusters coming from dependencies is smaller than the clusters coming from nouns.

5.2 Result of Cluster Labeling

We compared the label of the clusters between nouns and dependencies. We compared them for 10 clusters. Table 5 shows the number of the cluster, the number of the records (the number of documents) and the labels of each cluster, for noun.

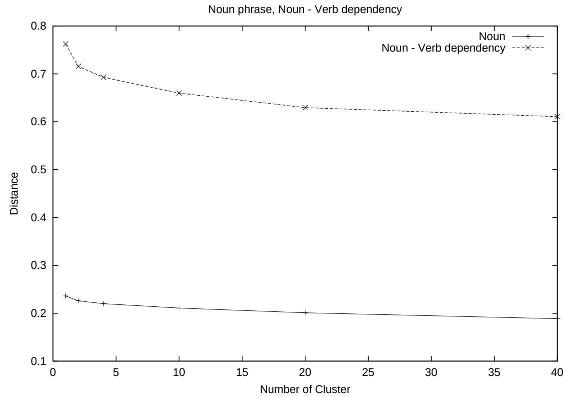

Figure 9: Average distance from centroid

Each label of a cluster is the terms in that cluster that are selected from the 20 most frequent terms in that cluster. Then, we eliminated two terms *"error"* and *"message"* (since they appeared in the most of the clusters), and eliminated symbols such as "." from these 20 terms. Up to 5 terms from the remaining terms are described in the Table 5. Some bracketed terms are specific operating system "*(OS)*" or products "*(Product)*" .

Table 6 shows the dependencies, using our methods. The method of selection of labels is the same as Table 5. ((*1) : *"oto ga kikoeru"* in Japanese is a dependency noun and verb, *"sound"* in English.)

Compared with two labeling results, for nouns, most of the clusters contain product names. However, it is hard to grasp the textual content only from the product names or accessory names.

In striking contrast to nouns, the result of dependencies shows that dependencies can easily capture the context of the original document set. For example, only viewing the labels, we can easily guess that cluster 3 is about memory problems, and cluster 7 is about sound problems.

6 Conclusion

Text mining aims not only at discovering knowledge, but also at showing an overview of a large collection of textual data. Since the mining results should be as informative as possible, we have proposed a schema using NLP for text mining significant terms and dependencies to ex-

Table 5: Clusters: Nouns

Clust	Records	Label
0	1530	PC, (OS1), screen, HDD, power supply
1	404	(Product1), (Product5), screen, modem, (OS1)
2	876	IBM, modem, setting, Internet, port
3	754	(Product2), setting, 535, Internet, screen
4	1373	(Product3), modem, 770, screen, status
5	1189	(OS1), power supply, screen, status, HDD
6	33909	screen, (Product1), HDD, (Product4), Internet
7	744	problem, status, screen, customer, setting
8	945	screen, modem, recovery, status, file
9	1386	phone, software, customer, CD-ROM. screen
Total	43110	

Table 6: Clusters: Dependencies

Clust	Records	Label
0	292	make..recovery, turn on..power, connect..Internet, not change..status, remove..option
1	440	turn on..power, turn off..power, not display..screen, freeze..screen
2	375	found..problem, hope..contact, not change..status, turn on..power
3	299	add..memory, add..memoryy, tell..service center, make..recovery
4	40189	turn off..power, connect..Internet, start..(OS1), install..(OS1)
5	235	not start..(OS1), push..key, start..(OS1), display..screen, not change..status
6	314	re-install..(OS1), not recognize..CD-ROM, can't start..(OS1), do..re-installation, happen..exception
7	456	sound..(*1) , want to know...method, not sound..(*1), not sound..speaker,
8	335	not change..status, make..recovery, get..call
9	175	can't connect..Internet, make..recovery, connect..Internet, have..modem
Total	43110	

clude trivial and rare content aggregations.

Our experiments showed that the significant terms, intentions and dependencies capture the textual content and improved the content aggregation much better than simple keywords.

7 Future Works

In this paper, we described the result of cluster labeling as a one of applications of text mining. We will further explore trend analysis and FAQ generation based on significant terms and dependencies.

And we will improve the function of our method. In order to extract domain-specific compound noun, we used the average of entropy value of the element terms. It is desirable to be normalized by the frequency.

References

Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. 1993. "Mining Association Rules between Sets of Items in Large Databases". *Proceedings of the 1993 International Conference on Management of Data (SIGMOD-93)*, pages 207–216.

Ryoichi Fujino and Hiroki Arimura. 2000. "Discovering Unordered and Ordered Phrase Association Patterns for Text Mining". *Pro-*

ceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-00), pages 281–293.

Ralph Grishman. 1997. *"Information Extraction: Techniques and Challenges"*. Lecture Notes in Artificial Intelligence. Springer Verlag.

Udo Hahn and Klemens Schnattinger. 1997. "Deep Knowledge Discovery from Natural Language Texts". *Proceedings of the 1997 International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 175–178.

Marti A Hearst. 1999. "Untangling Text Data Mining". *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 3–10.

Toru Hisamitsu, Yoshiki Niwa, and Jun ichi Tsujii. 2000. "A Method of Measuring Term Representativeness - Baseline Method Using Co-occurrence Distribution -". *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 320–326.

Kevin Knight. 1999. "Mining Online Text". *Communications of the ACM*, 42(11):58–61.

Hiroshi Maruyama and Shiho Ogino. 1994. "Japanese Morphological Analysis Based on Regular Grammar". *Transactions of Information Processing Society of Japan*, 35(7):1293–1299.

Hirofumi Matsuzawa and Takeshi Fukuda. 2000. "Mining Structured Association Patterns from Databases". *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-00)*, pages 233–244.

Dunja Mladenic. 1999. "Text-Learning and Related Intelligent Agent: A Survey". *IEEE Intelligent Systems*, 14(4):44–54.

Philip Resnik. 1995. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy". *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 448–453.

Gerald Salton. 1983. *"The SMART and SIRE Experimental Retrieval Systems"*. McGraw-Hill.

Tomek Strzalkowski and Barbara Vauthey. 1992. "Information Retrieval using Robust Natural Language Processing". *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*, pages 104–111.