Feature Selection in Categorizing Procedural Expressions

Mineki Takechi[*‡], **Takenobu Tokunaga**[†], **Yuji Matsumoto**[‡], **Hozumi Tanaka**[†]
[*]Fujitsu Limited
17-25 Shinkamata 1-chome, Ota-ku, Tokyo 144-8588, Japan
[†]Department of Computer Science, Tokyo Institute of Technology
2-12-2 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
[‡]Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma city, Nara 630-0101, Japan
{mineki-t,matsu}@is.aist-nara.ac.jp, {take,tanaka}@cl.cs.titech.ac.jp

Abstract

Text categorization, as an essential component of applications for user navigation on the World Wide Web using Question-Answering in Japanese, requires more effective features for the categorization of documents and the efficient acquisition of knowledge. In the questions addressed by such navigation, we focus on those questions for procedures and intend to clarify specification of the answers.

1 Introduction

Recent methodologies of text categorization as applied to Question-Answering(QA) and user navigation on the Web address new types of problems, such as the categorization of texts based on the question type in addition to one based on domain and genre. For good performance in a shallow approach, which exploits the shallow specification of texts to categorize them, requires a great deal of knowledge of the expressions in the answers corresponding to the questions. In most past QA research, the types of question have been primarily restricted to fact-based questions. However, in user navigation on the Web, other types of questions should be supported. In this paper, we focus on questions requiring a procedure asking for such navigation and intend to study the features necessary for its extraction by illustrating the specification of its answer. In the above type of QA, very few studies have aimed at answering questions by extracting *procedural expressions* from web pages. Accordingly, a) representations in a web text to indicate a procedure, b) the method of extracting those representations, and c) the way to combine related texts as an answer, are issues that have not been sufficiently clarified. Consequently, past studies do not provide a general approach for solving this task.

In contrast, it has been reported that the texts related to QA in web pages contain many lists in the descriptions. We decided to focus on lists including procedural expressions and employed an approach of extracting lists from web pages as answers. This results in difficulty in extracting the answers written in a different style. However, compared to seeking answer candidates from a document set including various web pages, it is expected that they will be found relatively more often from the gathered lists. In this study, our motivation is to provide users with the means to navigate accurately and credibly to information on the Web, but not to give a complete relevant document set with respect to user queries. In addition, a list is a summarization made by humans, and thus it is edited to make it easy to understand. Therefore, the restriction to itemized answers doesn't lose its effectiveness in our study. In the initial step of our work for this type of QA, we discuss a text categorization task that divides a set of lists into two groups: procedural and non-procedural. First, we gathered web pages from a search engine and extracted lists including the procedural expressions tagged with any HTML(Hyper Text Markup Language) list tags found, and observed their characteristics. Then we examined Support Vector Machines (SVMs) and sequential pattern mining relative to the set of lists, and observed the obtained model to find useful features for extraction of answers to explain a relevant procedure. In the following section, we introduce some related work. Section 3 presents the list features including procedural expressions in the web pages. Subsequently, we will apply our machine learning and sequential pattern mining techniques to learn these features, which are briefly illustrated in Section 4. Section 5 shows the results of our categorization experiments. Finally, Section 6 presents our conclusions and Section 7 gives our plans for future study.

2 Related Works

The questions related in all procedures were addressed by an expert system(Barr et al., 1989). However, in QA and information retrieval for open domain documents from the Web, the system requires a more flexible and more machine-operable approach because of the diversity and changeable nature of the information resources. Many competitions, e.g. TREC and NTCIR, are being held each year and various studies have been presented (Eguchi et al., 2003; Voorhees, 2001). Recently, the most successful approach has been to combine many shallow clues in the texts and occasionally in other linguistic resources. In this approach, the performance of passage retrieval and categorization is vital for the performance of the entire system. In particular, the productiveness of the knowledge of expressions corresponding to each question type, which is principally exploited in retrieval and categorization, is important. In this perspective, that means that the requirements for categorization in such applications are different from those in previous categorizations. Many studies have been made that are related to QA. Fujii et al.(2001) studied QA and knowledge acquisition for definition type questions. Approaches by seeking any answer text in the pages of FAQs or newsgroups appeared in some studies(Hamada et al., 2002; Lai et al., 2002). Automatic QA systems in a support center of organizations was addressed in a study by Kurohashi et al.(2000).

However, most of the previous studies targeting QA address fact type or definition type questions, such as ”When was Mozart born?” or ”What is platinum?”. Previous research addressing the type of QA relevant to procedures in Japanese is inconclusive. In text categorization research, the feature selection has been discussed(Taira and Haruno, 2000; Yang and Pedersen, 1997). However, most of the research addressed categorization into taxonomy related to domain and genre. The features that are used are primarily *content words*, such as nouns, verbs, and adjectives. Function words and frequent formative elements were usually eliminated. However, some particular areas of text categorization, for example, authorship identification, suggested a feasibility of text categorization with functional expressions on a different axis of document topics. From the perspective of seeking methods of domain-independent categorization for QA, this paper investigates the feasibility of *functional expressions* as a feature for the extraction of lists including procedural expressions.

Table 1: Result from a Search Engine.

Keyword	Gathered	Retrieved	Vaild Pages
tejun	3,713	748	629
houhou	5,998	916	929

Table 2: Domain and Type of List.

Domain	Procedures	Non-Procedures	All
Computer	558 (295)	1666 (724)	2224
Others	163 (64)	1733 (476)	1896
All	721	3399	4120

3 Extraction of Procedural Expressions

3.1 Answering Procedures with Lists

We can easily imagine a situation in which people ask procedural questions, for instance a user who wants to know the procedure for installing the RedHat Linux OS. When using a web search engine, the user could employ a keyword related to the domain, such as ”RedHat,” ”install,” or the synonyms of ”procedure,” such as ”method” or ”process.” In conclusion, the search engine will often return a result that does not include the actual procedures, for instance, only including the lists of hyperlinks to some URLs or simple alternatives that have no intentional order as is given.

This paper addresses the issue in the context of the solution being to return to the actual procedure. In the initial step of this study, we focused on the case that the continuous answer candidate passage is in the original text and furthermore restricted the form of documentation in the list. The list could be expected to contain important information, because it is a summarization done by a human. It has certain benefits pertaining to computer processing. These are: a) a large number of lists in FAQs or homepages on web pages, b) some clues before and after the lists such as title and leads, c) extraction which is relatively easy by using HTML list tags, e.g. `<OL>`,`<UL>`.

In this study, a binary categorization was conducted, which divided a set of lists into two classes of procedures and non-procedures. The purpose is to reveal an effective set of features to extract a list explaining the procedure by examining the results of the categorization.

3.2 Collection of Lists from Web Pages

To study the features of lists contained in web pages, the sets of lists were made according to the following steps (see Table 1) :

Step 1 Enter *tejun* (procedure) and *houhou* (method) to Google(Brin and Page, 1998) as keywords, and obtain a list of URLs that are to serve as the seeds of collection for the next step (*Gathered*).

Step 2 Recursively search from the top page to the next lower page in the hyperlink structure and gather the HTML pages (*Retrieved*).

Step 3 Extract the passages from the pages in Step 2 that are tagged with `<OL>` or `<UL>`. If a list has multiple layers with nested tags, each layer is decomposed as an independent list (*Valid Pages*).

Step 4 Collect lists including no less than two items. The document is created in such a way that an article is equal to a list.

Subsequently, the document set was categorized into procedure type and non-procedure type subsets by human judgment. For this categorization, the definition of the list to explain the procedure was as follows: a) The percentage of items including actions or operations in a list is more than or equal to 50%. b) The contexts before and after the lists are ignored in the judgment. An item means an article or an item that is prefixed by a number or a mark such as a bullet. That generally involves multiple sentences. In this categorization, two people categorized the same lists and a kappa test(Siegel and Castellan, 1988) is applied to the result. We obtained a kappa value of 0.87, i.e., a near-perfect match, in the computer domain and 0.66, i.e., a substantial match, in the other domains. Next, the documents were categorized according to their domain by referring to the page including a list. Table 2 lists the results. The values in parentheses indicate the number of lists before decomposition of nested tags. The documents of the *Computer* domain were dominant; those of the other domains consisted of only a few documents and were lumped together into a document set named "*Others*." This domain consists of documents regarding education, medical treatment, weddings, etc. The instructions of software usage or operation on the home pages of web services were also assigned to the computer domain.

3.3 Procedural Expressions in the Lists

From the observations of the categorized lists made by humans, the following results were obtained: a) The first sentence in an item often describes an action or an operation. b) There are two types of items that terminate the first sentence: nominalized and nonnominalized. c) In the case of the nominalized type, verbal nouns are very often used at the end of sentence. d) Arguments marked by *ga* (a particle marking nominative) or *ha* (a particle marking topic) and negatives are rarely used, while arguments marked by *wo* (a particle marking object) appear frequently. e) At the end of sentences and immediately before punctuation marks, the same expressions appear repeatedly. Verbal nouns are inherent expressions verbified by being followed by the light verb *suru* in Japanese. If the features above are domain-independent characteristics, the lists in a minor domain can be categorized by using the features that were learned from the lists in the other major domain. The function words or flections appearing at the ends of sentences and before punctuation are known as markers, and specify the style of descrip-

Table 3: Types of Tags.

	tag type	object types
Document	*dv*	list
	p	item
	su	sentence
Part of Speech	*np*	noun[1] prefix
	snp	verbal noun
	vp	verb
	adp	particle[2] adverb adnominal conjunction
	ajp	adjective
	aup	sentece-final-particle auxiliary verb suffix
	ij	interjection
	seg	others (punctuation, etc.)
	unknown	unknown word

tion in Japanese. Thus, to explain a procedure, the list can be expected to have inherent styles of description.

These features are very similar to those in an authorship identification task(Mingzhe, 2002; Tsuboi and Matsumoto, 2002). That task uses word n-gram, distribution of part of speech, etc. In recent research for web documents, frequent word sequences have also been examined. Our approach is based on these features.

4 Features

4.1 Baseline

In addition to the features based on the presence of specific words, we examined sequences of words for our task. Tsuboi et al.(2002) used a method of *sequential pattern mining*, PrefixSpan, and an algorithm of machine learning, Support Vector Machine in addition to morphological N-grams. They proposed making use of the frequent sequential patterns of words in sentences. This approach is expected to contribute to explicitly use the relationships of distant words in the categorization. The list contains differences in the omissions of certain particles and the frequency of a particle's usage to determine whether the list is procedural. Such sequential patterns are anticipated to improve the accuracy of categorization. The words in a sentence are transferred to PrefixSpan after preprocessing, as follows:

Step 1 By using ChaSen(Matsumoto et al., 1999), a Japanese POS(Part Of Speech) tagger, we put the document tags and the POS tags into the list. Table 3 lists the tag set that was used. These tags are only used for distinguishing objects. The string of tags was ignored in sequential pattern mining.

Step 2 After the first n sentences are extracted from each list item, a sequence is made for each sentence. Sequential pattern mining is performed for an item (literal) in a sequence as a morpheme.

By using these features, we conducted categorization with SVM. It is one of the large margin classifiers, which shows high generalization performance even in high dimensional spaces(Vapnik, 1995). SVM is beneficial for our task, because it is unknown which features are effective, and we must use many features in categorization to investigate their effectiveness. The dimension of the feature space is relatively high.

4.2 Sequential Pattern Mining

Sequential pattern mining consists of finding all frequent subsequences, that are called *sequential patterns*, in the database of sequences of literals. Apriori(Agrawal and Srikant, 1994) and PrefixSpan(Pei et al., 2001) are examples of sequential pattern mining methods. The Apriori algorithm is one of the most widely used methods, however there is a great deal of room for improvement in terms of calculation cost. The PrefixSpan algorithm succeed in reducing the cost of calculation by performing an operation, called *projection*, which confines the range of the search to sets of frequent subsequences. Details of the PrefixSpan algorithm are provided in another paper(Pei et al., 2001).

[1] Except verbal nouns

[2] Except sentence-final particles

Table 4: Statistics of Data Sets.

	Proc.	Non-Proc.	*Comp.*	*Others*
Lists	721	3399	2224	1896
Items	4.6 / 2.8	4.9 / 5.7	4.8 / 6.1	4.9 / 4.4
Sen.	1.8 / 1.7	1.3 / 0.9	1.5 / 1.1	1.3 / 1.1
Char.	40.3 / 48.6	32.6 / 42.4	35.6 / 40.1	32.6 / 48.2

5 Experiments and Results

5.1 Experimental Settings

In the first experiment, to determine the categorization capability of a domain, we employed a set of lists in the *Computer* domain and conducted a cross-validation procedure. The document set was divided into five subsets of nearly equal size, and five different SVMs, the training sets of four of the subsets, and the remaining one classified for testing. In the second experiment, to determine the categorization capability of an open domain, we employed a set of lists from the *Others* domain with the document set in the first experiment. Then, the set of the lists from the *Others* domain was used in the test and the one from the *Computer* domain was used in the training, and their training and testing roles were also switched. In both experiments, recall, precision, and, occasionally, F-measure value were calculated to evaluate categorization performance. F-measure is calculated with precision (P) and recall (R) in formula 1.

$$F = \frac{2PR}{P+R} \tag{1}$$

The lists in the experiment were gathered from those marked by the list tags in the pages. To focus on the feasibility of the features in the lists for the categorization task, the contexts before and after each list are not targeted. Table 4 lists four groups divided by procedure and domain into columns, and the numbers of lists, items, sentences, and characters in each group are in the respective rows. The two values in each cell in Table 4 are the mean on the left and the deviation on the right. We employed Tiny-SVM[1] and a implementation of PrefixSpan[2] by T. Kudo. To observe the direct effect of the features, the feature vectors were binary, constructed with word N-gram and patterns; polynomial kernel degree d for the SVM was equal to one. Support values for PrefixSpan were determined in an ad hoc manner to produce a sufficient number of patterns in our experimental conditions.

Table 5: POS Groups.

	Combination of POS	*Computer*	*Others*
F1	all of words	9885	13031
F2	snp+np+vp+ajp	4570	7818
F3	snp+np+vp+ajp+unknown	9277	12169
F4	aup+adp+seg	608	862
F5	aup+adp+seg+unknown	5315	5213
F6	snp+aup+adp+seg	1493	2360

To investigate the effective features for list categorization, feature sets of the lists were divided into five groups (see Table 5) with consideration given to the difference of content word and function words according to our observations (described in Section 3.3). The values in Table 5 indicate the numbers of differences between words in each domain data set. The notation of tags above, such as 'snp', follows the categories in Table 3. F2 and F3 consist of content words and F4 and F5 consist of function words. F6 was a feature group, which added verbal nouns based on our observations (described in Section 3.3).

To observe the performances of SVM, we compared the results of categorizations in the conditions of F3 and F5 with a decision tree. For decision tree learning, j48.j48, which is an implementation of the C4.5 algorithm by Weka[3], was chosen.

In these experiments, only the first sentence in each list item was used because in our preliminary experiments, we obtained the best results when only the first sentence was used in categorization. As many as a thousand patterns from the top in the ranking of frequencies were selected and used in conditions from F1 to F6. For pattern selection, we examined the method based on frequency. In addition, mutual information filtering was conducted in some conditions for comparison with performances based only on pattern frequency. By ranking these with the mutual information filtering, we selected 100, 300,

[1] http://cl.aist-nara.ac.jp/˜taku-ku/software/TinySVM/

[2] http://cl.aist-nara.ac.jp/˜taku-ku/software/prefixspan/

[3] http://www.cs.waikato.ac.nz/˜ml/weka/

Table 6: Result of Close-Domain.

Computer domain				
	1	1+2	1+2+3	pattern
F1	0.88/0.88	0.92/0.90	0.93/0.90	0.93/0.92
F2	0.85/0.86	0.90/0.87	0.91/0.85	0.89/0.88
F3	0.87/0.86	0.93/0.87	0.93/0.86	0.91/0.88
F4	0.81/0.81	0.85/0.85	0.86/0.86	0.86/0.86
F5	0.81/0.84	0.86/0.85	0.90/0.86	0.89/0.88
F6	0.85/0.87	0.90/0.89	0.91/0.89	0.89/0.89

Table 7: Results when Learning from *Computer* Domain.

Computer Domain - *Others* Domain				
	1	1+2	1+2+3	pattern
F1	0.60/0.46	0.69/0.45	0.72/0.45	0.66/0.48
F2	0.52/0.42	0.69/0.39	0.72/0.37	0.64/0.41
F3	0.56/0.46	0.68/0.44	0.70/0.42	0.63/0.45
F4	0.46/0.51	0.59/0.58	0.58/0.52	0.53/0.60
F5	0.43/0.50	0.52/0.48	0.61/0.48	0.53/0.53
F6	0.53/0.49	0.67/0.53	0.71/0.50	0.61/0.55

Table 8: Results when Learning from *Others* Domain.

Others Domain - *Computer* Domain				
	1	1+2	1+2+3	pattern
F1	0.90/0.52	0.95/0.60	0.97/0.56	0.95/0.64
F2	0.88/0.51	0.92/0.44	0.94/0.37	0.94/0.47
F3	0.90/0.46	0.95/0.48	0.97/0.41	0.96/0.49
F4	0.80/0.33	0.79/0.58	0.79/0.55	0.79/0.59
F5	0.83/0.51	0.85/0.54	0.88/0.51	0.87/0.53
F6	0.81/0.51	0.90/0.56	0.94/0.51	0.89/0.56

and 500 patterns from 1000 patterns. Furthermore, the features of N-grams were varied to N=1, 1+2, and 1+2+3 by incrementing N and adding new N-grams to the features in the experiments.

5.2 Experimental Results

Table 6 lists the results of a 5-fold cross-validation evaluation of the *Computer* domain lists. Gradually, N-grams and patterns were added to input feature vectors, thus N=1, 2, 3, and patterns. The feature group primarily constructed of content words slightly overtook the function group, with the exception of recall, while trigram and patterns were added. In the comparison of F2 and F4, differences in performance are not as salient as differences in numbers of features. Incorporating verbal nouns into the categorization slightly improved the results. However, the patterns didn't work in this task. The same experiment-switching the roles of the two list sets, the *Computer* and the *Others* domain, was then performed (see Tables 7 and 8).

Along with adding N-grams, the recall became worse for the group of content words. In contrast, the group of function words showed better performance in the recall, and the overall balance of precision and recall were well-performed. Calculating the F-measure with formula 1, in most evaluations of open domain, the functional group overtook the content group. This deviation is more salient in the *Others* domain. In the results of both the *Computer* domain and the *Others* domain, the model trained with functions performed better than the model trained with content. The function words in Japanese characterize the descriptive style of the text, meaning that this result shows a possibility of the acquisition of various procedural expressions. From another perspective, when trigram was added as a feature, performance took decreased in recall. Adding the patterns, however, improved performance. It is assumed that there are dependencies between words at a distance greater than three words, which is beneficial in their categorization. Table 9 compares the results of SVM and j48.j48 decision tree. Table 10 lists the effectiveness of mutual information filtering. In both tables, values show the F-measure calculated with formula 1. According to Table 9, SVM overtook j48.j48 overall. j48.j48 scarcely changes with an increase in the number of features, however, SVM gradually improves performance. For mutual information filtering, SVM marked the best results with no-filter in the *Computer* domain. However, in the case of learning from the *Others* domain, the mutual information filtering appears effective.

5.3 Discussion

The comparison of SVM and decision tree shows the high degree of generalization of SVM in a high dimensional feature space. From the results of mutual information filtering, we can recognize that the sim-

Table 9: Comparison of SVM and Decision Tree.

	1		1+2		1+2+3		
	SVM	j48	SVM	j48	SVM	j48	#feature
F3	0.84	0.79	0.84	0.83	0.84	0.83	300
	0.85	0.76	0.85	0.81	0.84	0.82	500
	0.84	0.76	0.86	0.82	0.86	0.83	1000
	0.87	0.76	0.87	0.82	0.87	0.83	5000
F5	0.84	0.79	0.84	0.82	0.82	0.81	300
	0.85	0.80	0.85	0.81	0.83	0.82	500
	0.86	0.80	0.86	0.81	0.84	0.81	1000
	0.84	0.80	0.86	0.82	0.87	0.82	5000

Table 10: Results of Pattern Selection with Mutual Information Filtering.

		100	300	500	no-filter
Computer	*F3*	0.53	0.53	0.53	0.52
- *Others*	*F5*	0.53	0.52	0.50	0.53
Others	*F3*	0.74	0.74	0.75	0.65
- *Computer*	*F5*	0.75	0.76	0.77	0.66

ple methods of other pre-cleaning are not notably effective when learning from documents of the same domain. However, the simple methods work well in our task when learning from documents consisting of a variety of domains.

Patterns performed well with mutual information filtering in a data set including different domains and genres. It appears that N-grams and credible patterns are effective in acquiring the common characteristics of procedural expressions across different domains. There is a possibility that the patterns are effective for moderate narrowing of the range of answer candidates in the early process of QA and Web information retrieval. In the *Computer* domain, categorization performed well overall in every POS group. That is why it includes many instruction documents, for instance software installation, computer settings, online shopping, etc., and those usually use similar and restricted vocabularies. Conversely, the uniformity of procedural expressions in the *Computer* domain causes poorer performance when learning from the documents of the *Computer* domain than when learning from the *Others* domain. We also often found in their expressions that for a

Sentence : “ [*menyu*] *wo sentaku shi,*
“ Select [menu] and
[*hozon*] *wo kurikku suru* . ”
click the switch of [save] . ”

Pattern 1 : ‘[’ ‘]’ ‘*wo*’ ‘,’
Pattern 2 : ‘[’ ‘]’ ‘*wo*’ ‘.’

Figure 1: Example of Effective Patterns.

particular class of content word, special characters were adjusted (see Figure 1). This type of pattern occasionally contributed the correct classification in our experiment. The movement of the performance of content and function word along with the addition of N-grams is notable. It is likely that making use of the difference of their movement more directly is useful in the categorization of procedural text.

By error analysis, the following patterns were obtained: those that reflected common expressions, including the multiple appearance of verbs with a case-marking particle *wo*. This worked well for the case in which the procedural statement partially occupied the items of the list. Where there were fewer characters in a list and failing POS tagging, pattern mismatch was observed.

6 Conclusion

The present work has demonstrated effective features that can be used to categorize lists in web pages by whether they explain a procedure. We show that categorization to extract texts including procedural expressions is different from traditional text categorization tasks with respect to the features and behaviors related to co-occurrences of words. We also show the possibility of filtering to extract lists including procedural expressions in different domains by exploiting those features that primarily consist of function words and patterns with mutual information filtering. Lists with procedural expressions in the *Computer* domain can be extracted with higher accuracy.

7 Future works

The augmentation of the volume of data sets within the *Others* domain is a considerable task. In this research, the number of lists in each specific domain of the data set within the *Others* domain is too few to reveal its precise nature. In more technical domains, the categorization of lists by humans is difficult for people who have no knowledge of the field. Another unresolved problem is the nested structure of lists. In our current method, no list is nested because it has already been decomposed during preprocessing. In some cases, this treatment incorrectly categorizes lists that can be regarded as procedural types into another group based on the condition of accepting a combination of two or more different layers of nested lists. Another difficult point is related to the nominal list type. According to the observations of the differences in categorization in the *Others* domain by humans, some failures are of the nominal type. It is difficult to distinguish such cases by features only in lists, and more clues to recognize the type of list are required such as, for example, the contexts before and after the list.

Acknowledgements

My deepest gratitude is to Taku Kudo who provided Tiny-SVM and an implementation of PrefixSpan.

References

Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rulesr. In *Proceedings of 20th International. Conference. Very Large Data Bases (VLDB)*, pages 487–499.

A. Barr, P. R. Cohen, and E. A. Feigenbaum. 1989. *The Handbook of Artificial Intelligence*. Kyoritsu Shuppan, Tokyo. Japanese Edition Translated by K. Tanaka and K. Fuchi.

S. Brin and L. Page. 1998. The Anatomy of a Large-Scale Hypertexual Web Search Engine. In *Proceedings of 7th International World Wide Web Conference*.

Koji Eguchi, Keizo Oyama, Emi Ishida, Noriko Kando, and Kazuko Kuriyama. 2003. Overview of the Web Retrieval Task at the Third NTCIR Workshop. Technical Report NII-2003-002E, National Institute of Informatics.

Atsushi Fujii and Tetsuya Ishikawa. 2001. Organizing Encyclopedic Knowledge based on the Web and its Application to Question Answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-EACL 2001)*, pages 196–203, July.

Reiko Hamada, Ichiro Ide, Shuichi Sakai, and Hidehiko Tanaka. 2002. Structural Analysis of Cooking Preparation Steps. *The Transactions of The Institute of Electronics*, D-II Vol.J85-D-II(1):79–89, January. (in Japanese).

Sadao Kurohashi and Wataru Higasa. 2000. Dialogue Helpsystem based on Flexible Matching of User Query with Natural Language Knowledge Base. In *Proceedings of 1st ACL SIGdial Workshop on Discourse and Dialogue*, pages 141–149.

Y. Lai, K. Fung, and C. Wu. 2002. FAQ Mining via List Detection. In *Proceedings of Workshop on Multilingual Summarization and Question Answering (COLING)*.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, and Tomoaki Imamura. 1999. Japanese Morphological analysis System ChaSen Manual. Naist Technical Report NAIST-IS-TR99009, Nara Institute of Science and Technology. (in Japanese).

Jin Mingzhe. 2002. Authorship Attribution Based on N-gram Models in Postpositional Particle of Japanese. *Mathematical Linguistic*, 23(5):225–240, June.

Jian Pei, Jiawei Han, et al. 2001. Prefixspan: Mining Sequential Patterns by Prefix-Projected Growth. In *Proceedings of International Conference of Data Engineering*, pages 215–224.

S. Siegel and NJ. Castellan, Jr. 1988. *Nonparametric Statistics for the Behavioral Sciences 2nd Edition*. McGraw-Hill, New York.

Hirotoshi Taira and Masahiko Haruno. 2000. Feature Selection in SVM Text Categorization. *IPSJ Journal*, 41(4):1113–1123, April. (in Japanese).

Yuta Tsuboi and Yuji Matsumoto. 2002. Authorship Identification for Heterogeneous Documents. In *IPSJ SIG Notes*, NL-148-3, pages 17–24.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York.

Ellen M. Voorhees. 2001. Overview of the TREC 2001Question Answering Track. In *Proceedings of the 2001 Text Retrieval Conference (TREC 2001)*.

Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of ICML-97 14th International Conference on Machine Learning*, pages 412–420.