NAIST-IS-DD0061208

Doctoral Dissertation

Identification of Multi-Sentence Question Type and Extraction of Descriptive Answer in Open Domain Question-Answering

Mineki Takechi

March 19, 2007

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Mineki Takechi

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Shunsuke Uemura	(Member)
Associate Professor Takenobu Tokunaga	(Member, TITECH)
Associate Professor Kentaro Inui	(Member)

Identification of Multi-Sentence Question Type and Extraction of Descriptive Answer in Open Domain Question-Answering*

Mineki Takechi

Abstract

The state-of-art question-answering systems provide the answer to the user by directly extracting the exact answer from a huge amount of documents in large databases or Web pages. On the other hand, many online question-answering services, such as automatic answering services in call centers, Q&A web sites in the Internet, are in operation by restricting the domain of questions, and by utilizing manpower to provide the answers. Characteristics of queries handled in these two kinds of systems are different in terms of their length of queries and the type of questions in the queries. The queries of advanced question-answering are usually represented as a single sentence mainly consisting of a factoid question item. Factoid question can be usually answered by a noun phrase, such as a person name, an organization name, a location name, and so on. In contrast, the queries in a practical question-answering service often consist of multiple sentences comprising multiple non-factoid question items that ask a definition, a procedure, an opinion, and so forth. In numerous cases, they require descriptive answers. Despite long history of studies in question-answering, advanced question-answering has not sufficiently supported non-factoid multi-sentence question yet. This thesis aims to develop an open domain question-answering system that can address multi-sentence queries requiring descriptive answers.

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0061208, March 19, 2007.

This thesis studies two fundamental components for realizing this kind of question-answering, that are, domain-independent multi-sentence question analysis and descriptive answer extraction from large amount of documents. In question analysis, this thesis examines methodologies to decompose a multi-sentence query to question items and identify their question types. To achieve these tasks, this thesis proposes a new efficient sentence-chunking based technique. In addition to conventional single sentence questions, the proposed method can handle a multi-sentence query comprising multiple question items, as well as a question item comprising multiple sentences in the same framework.

Additionally, this thesis studies the typology of descriptive answers in Q&A services and the automatic categorization of descriptive answers in terms of the typology. Exploiting features of functional words that have not been counted in the previous work of text categorization, this study proposes new techniques of answer categorization based on the description type. The experimental results showed a high accuracy of the proposed method, 0.8 F-measure, for the three description types: definition, chronological order and procedure.

Moreover, this thesis pays attention to the answer extraction from web pages. Our approach accurately extracts answer candidates by identifying the description type. The new method is based on sequential pattern mining and machine learning techniques to extract lists of procedural expressions. As a result of experiments, this method showed a high performance, more than 0.7 F-measure, when extracting lists of procedural expressions.

Keywords:

question-answering, multiple sentence question, non-factoid question, descriptive answer, Web question-answering, procedural expressions

分野を限定しない質問応答における複数文質問の識別と記述的な回答の抽出*

武智 峰樹

内容梗概

近年、インターネットやデータベース中にある大量の文書から、ユーザの質問に対する回答を直接抽出する質問応答の研究が盛んに行われている。そうした質問応答では、ユーザは分野に関わらず質問を行うことができる。一方、産業分野では、電話回線を通じたコールセンターなどのサービスや、インターネット上でのオンライン質問応答サービスが数多く存在している。このようなサービスでは、扱う質問を特定の分野に限定したり、人手による回答を行うなど実用的な手段で質問応答が運用されている。この２つの質問応答が扱う質問は２つの点で異なる。前者で扱う質問は、人名、組織名、場所など名詞で回答できる単一の文からなる質問であることが多い。一方、実際の質問応答サービスが扱う質問は、複数の文を含み、定義、方法、意見など回答として比較的長い記述を必要とするタイプの質問が少なくない。こうした記述的な回答を必要とする質問については、分野を限定しない質問応答では十分に扱うことができなかった。

本論文は、このような質問応答を行うために必要となる２つの要素技術について研究をおこなった。１つ目は、複数の文からなる質問を解析する技術であり、２つ目は大量の文書集合から記述的な回答を抽出する技術である。

複数の文からなる質問の解析では、従来手法に比べて効率的な文チャンキングに基づく新しい質問タイプ識別手法を提案した。提案手法は、複数の文からなる質問に含まれる質問事項を抽出し、その質問事項のタイプを識別することができ、これにより、従来研究で扱われてきた単一文からなる質問に加えて、複数の質問事項を含んでいる質問や、１つの質問事項が複数の文にわたる場合についても、同じ枠組みによって扱うことができるようになった。

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DD0061208, 2007年3月19日.

一方、回答抽出では、質問応答における記述的な回答の類型化とその自動分類技術について研究をおこなった。特に、機能語を中心とした素性を用いて、回答の記述タイプを分類する新しいテキスト分類について調査した。この結果、定義、時間的順序、手順の３つの記述タイプの分類については、F 値 0.8 を超える高い精度を得た。

さらに本論文では、インターネットのページから、回答を抽出する技術についても研究をおこなった。特に、記述タイプが手順タイプであるような箇条書きを抽出するタスクを取り上げ、シーケンシャルパターンマイニングと機械学習に基づく新しい手法を提案した。その結果、F 値 0.7 を超える高い精度により手順に関する箇条書きが抽出できることを示した。

キーワード

質問応答, 複数文質問, 非事実型質問, 記述的な回答, ウェブ質問応答, 手順

Contents

List of Tables

List of Figures

Acknowledgements

First, I would like to thank my supervisor, Professor Yuji Matsumoto: He was a role model in many aspects, in particular regarding my approach to research. His research and educational management of laboratory are quite suggestive and they always give me unexpected lessons for academic life. I am glad to have spent years as student at his laboratory.

Associate Professor Takenobu Tokunaga in the Tokyo Institute of Technology taught me the value of direct and simple thoughts. I learned the necessity of persistence and academic training in materializing the idea from his ingenuity and preciseness when he always shows his idea. His support was vital to finish my thesis. I would like to express my sincere thanks and appreciation for him.

Former Professor Hozumi Tanaka in the Tokyo Institute of Technology always encouraged me to keep studying and gave me the best environment for academic life. Without his exceptional understanding and generosity, I would not have been able to continue my study.

Associate Professor Kentaro Inui taught me the importance of careful linguistic observation through his studies in the era of statistical approach. Additionally, his flexible and broadminded spirit made my eyes opened to astonishing essence to keep motivating to research topics.

Thanks also to my committee members and advisors: Professor Shunsuke Uemura for reading my thesis and providing valuable comments; former Assistant Takashi Miyata and Edson T. Miyamoto, Assistant Masashi Shimbo and Masayuki Asahara, and Associate Professor Taiichi Hashimoto in the Tokyo Institute of Technology for valuable advices and thoughtful cares; Professors Akira Ichikawa and Atsushi Imiya in the Chiba University and Professor Takenori Makino in Toho University, who introduce insightful advices in the initial stage of this research.

The one person to whom I am indebted in bringing this thesis to completion is Shinichi Kubota. His constant encouragement and support, infinite patience and understanding, and willingness to let me establish my own professional ability, have all contributed to my humble development as a practitioner. Thanks go also to my other directors in Fujitsu who continue to support my studies at all times, Takao Fujimori, Hiroyuki Endo and Hitoshi Wada.

Kunio Matsui is also an individual to have had the influence on the development of this research. He was instrumental in giving me access to their Fujitsu Laboratories system resources, and also to other members of Text Information Processing Group, Hiroshi Tsuda, Fumihito Nishino, Kanji Uchino, Minako Hashimoto, Yoshio Nakao, and Hoshiai Tadashi. Yasuyo Kikuta and Sachiko Motoi provided tools and data for my research with members in Fujitsu Laboratories. I am grateful for their openness in sharing research ideas and resources.

A vast number of people have contributed both directly and indirectly to the development of this study. I would like to thank all the people.

Taku Kudo also made key contributions to this research in providing his excellent open softwares. His dedication to research has inspired me. Syoichi Kuboyama's tireless efforts in production of software realizing new ideas are my model. He also provided implementations of algorithms in this thesis. Tetsuro Takahashi willingly provided and enhanced his annotation tool for my studies. I appreciate his help.

My colleague Manabu Sassano always gave me friendship and support. His valuable comments as a senior NLP researcher really kept encouraging me. I would like to thank Akira Adachi. He also gave me valuable advices for academic life.

I am thankful to past and present members of NLP groups in TITECH and NAIST. I had many inspiring conversations with Akira Terada, Kotaro Funakoshi, Tatsumi Kobayashi, Kazuhiro Takeuchi, and Ryu Iida. Nozomi Kobayashi gave me valuable comments and help in formatting this thesis. Philipp Spanger, Masaki Noguchi, Daichi Kobayashi, Keita Hakoda, Taichi Watanabe, and Kayo Yamashita gave me great help. I appreciate their help in formatting and proofreading this text and annotating test datasets.

Close to home, without the support, occasional reprimand from Miwa Hamada, I could never get through the most painful years in my life.

I want to thank my family for their love and support, especially my parents Toru Takechi and Mitsue Takechi, my children Kumito and Otoha, my sister Yukiko Tominaga and my grandfather Tadao Kinoshita. My wife of blessed memory Emiko, without your encouragement in my mind I would never have began, and much less completed this thesis.

Chapter 1

Introduction

1.1 Motivation

Question-answering(QA) is the most natural way of exchanging information in human interaction. It is an ideal form in studies of question-answering in human-computer interaction, much of the studies have been gradually conducted to this goal. In a recent decade, the studies of new information accessibility to huge amount of documents have been made by TREC (Text Retrieval Conference) [82, 155] and NTCIR (NII Test Collection for IR Systems). Their studies comprise many advanced information access methodologies, such as speech interface framework, web information retrieval, information navigation, intelligent information access, and cross-language retrieval.

Current advanced question-answering can also be positioned in this stream of research, QA-Track in TREC [154] started in 1999 and NTCIR QAC(Question and Answering Challenge) [68] has been held annually since 2001. Question-answering work as a useful interface of an information retrieval engine that is able to accept sentence queries. Question-answering provides the answer to the user question by extracting the exact answer in retrieved articles from a large amount of source documents, databases or Web pages. Because this kind of question-answering addresses queries in unrestricted domains, it is called open domain question-answering. It mainly accepts only single sentence queries. For instance, questions in TREC take the form “What is the capital of Uruguay?” “How did Socrates die?” “Where is the Taj Mahal?” “When did the Jurassic

Period end? ", etc.

On the other hand, many online question-answering services, such as automatic answering services, call centers, helpdesk and Q&A sites, have already been established on the Internet and telephone networks. Q&A sites edit various questions and answers in FAQ style presentation. The queries are not restricted to certain domains, and the answers are written by the general public. Contrary to the queries in TREC and NTCIR, their queries comprise many multiple sentence queries.

There are two major differences between queries in TREC and in actual QA Services. The first difference is the length of the queries and the number of the sentences, and the second difference is the question type. A TREC query is basically a single sentence query (SSQ). Whereas a query in a QA service is often a multiple sentence query (MSQ). The questions of TREC require mainly factoid answers, such as a Person, an Organization and Location, but QA services must handle many non-factoid questions requiring descriptive answers that consist of a sentence or more, such as a Definition, a How-to and an Opinion, in addition to factoid questions. Although QA services have been supported by non-factoid MSQ, answers are extracted by humans. Figure 1.1 shows division of question-answering segmented by query, question and answer types in a tabular form. The queries in TREC can be assigned in the lower part of this table for factiod questions, and that in actual Q&A services includes divisions in the upper part of this table. In the upperhalf of this table, this thesis mainly deals with queries consisting of multiple sentences and non-factoid questions requiring answers by a sentence or more. In this thesis, this type of answer is called *descriptive answer*. To advance technologies of open domain question-answering to one that can apply more expanded query types such as those appearing in actual QA services, we have to develop open domain question-answering that can deal with multi-sentence queries and that handle questions requiring descriptive answers.

1.2 Focus of this research

In this thesis, we focus on two essential components to realize open domain multi-sentence question-answering. The first focus is *Question Segment Extraction* and

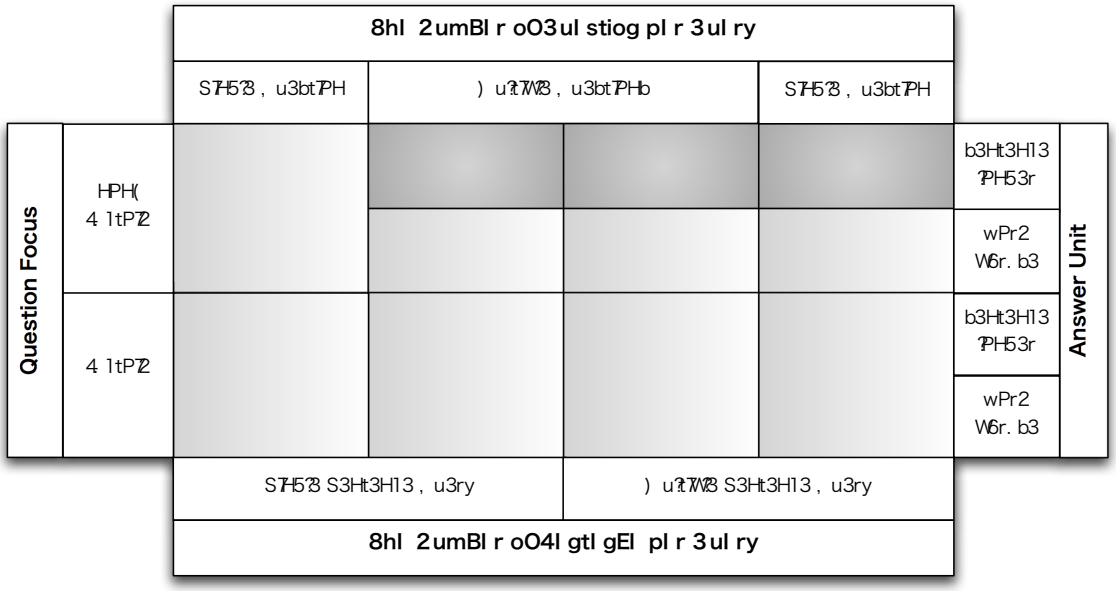

Figure 1.1. Division of Queries and Answer Types.

Question Type Identification for multi-sentence queries. The second is extraction of descriptive answers from Web pages toward question-answering requiring descriptive answers. We describe these two focuses in detail below.

Question Segment Extraction and Question Type Identification

In what way is a multi-sentence query different from a single sentence query? If a multi-sentence query is merely a set of SSQs, known question type identification methods would be directly applicable. A multi-sentence query can contain multiple questions. To avoid misunderstanding, each question included in a query is called a *question item* in this thesis as appropriate. Using this term, we firstly have to know how a query is decomposed into question items, and what context of a question item should be considered to classify the question item into either of the question types. Here, we call a set of sentences necessarily required to identify a question item and its type *question segment*. We also call the process by which a multi-sentence query is separated into question segments and then identify their question type of segments as question type identification. In this thesis, we clarify the structure of question segments and the conditions of ques-

tion type identification.

Extraction of Descriptive Answer

Achieving a descriptive answer required in question-answering (DQA) poses many difficulties. Examples of descriptive answers (DAs) include a How-to, a Condition, a Definition, an Opinion, a Reason and so forth. These answer types define types of questions. How do we extract these answers from their source articles? Firstly, we have to determine the DA boundary in a source article, *Answer Segment*. Secondly, we have to set various parameters to select relevant answers for the user query from variants of correct answers, such as fineness and concreteness of description, coverage of related information, degree of cross-reference between related documents, required document structure, subjectivity, or credibility, based on experience or speculation. Even if we suitably establish these conditions, we can consider multiple relevant answers according to the discourse structures in their answers. For instance, when we examine "Cut, boil and fill a bowl." is this a mere list of actions or a procedure? To deal with this type of problem correctly, we have to be able to recognize discourse relations, including, logical relations: parallelism, causality, supposition; temporal relation such as the order of actions, spatial relations such as the role and location of the agent, rhetorical relations such as exemplification and definition. Simple 'bag of words' features are insufficient for extracting the exact answer. Unfortunately, by current natural language processing (NLP), it is too difficult to solve all these problems.

There are two possible alternatives of condition setting of DQA. The first one is a restriction of a specific domain, such as cooking recipe [40, 121]. The second is restriction on the style of answers [13, 30, 31]. In some cases, we can exploit the style of description frequently appearing in an answer type to narrow down answer candidates. For instance, if we wish to know the meaning of *Soba*, "蕎麦", the answer style could mimic the description style of a dictionary, such as "Soba : Thin Japanese noodles made from buckwheat flour." Therefore, if we make preparations beforehand regarding the lexical and semantic patterns and then match the patterns to answer candidates, there are fewer and more relevant answer candidates to sort through. If we could also find a style that is dominant in a descriptive answer type, the style would possibly work well to identify correct

answers. Although different distributions of description style regarding different domains are predictable, some style can be considered to appear in various domains. Thus we can expect the feature of style in one domain to be also effective in other domains. What styles are frequently used in descriptive answers? How should a style of description, that is *description type*, be defined? Because we aim at extraction of answers from articles in documents, do we have to take account of linguistic expressions to define types of description style? Description type is not equal to general document style or format but are not individual writing style either. We intend to find description types that can be used to accurately extract each type of answer.

As another solution to the difficulty of DQA, we could take account of exploiting human annotated semantic meta-data in the case of difficulty in extracting the answer only using NLP, such as the example of a list and the procedure mentioned above. What style in a Q&A corpus can be annotated as semantic meta-data with high inter-annotator agreement? As the first step toward solving this problem, we performed description style annotation for Q&A articles and studied the annotation results, clarifying features of the description style of the answer. Using the features of style, we conducted experiments of extracting articles of a descriptive answer type, that is *procedural expression* from the Web pages. Additionally, we explored the effective features of the extraction of procedural expressions.

1.3 Guide to remaining chapters

We overview previous studies of question-answering and related researches in Chapter 2. Chapter 3 looks at multiple sentence query processing, and focuses on question segment extraction and question type identification for multi-sentence queries. Chapter 4 and Chapter 5 are devoted to answer extraction. We discuss annotation of description type to Q&A corpus in Chapter 4, and explore some expected description type resulted in annotation experiment. In Chapter 5, we propose the methodology of extraction of procedural expression from the Web pages using description type and machine learning, and show the effectiveness of the approach. Finally the thesis is concluded in Chapter 6.

Chapter 2

Concept and Architecture of Target Question-Answering

2.1 Introduction

From pioneer works of artificial intelligence in 1960s [9, 28, 72, 157, 158] to open-domain question-answering researches in natural language processing and information retrieval as typified by TREC or NTCIR [26, 32–34, 66, 82, 154, 155], many types of question-answering systems have been proposed. The question-answering is performed in varied environments, as well as the task and role of proposed systems in their environments. The requirements of question-answering systems can be considered in three viewpoints related to the environment, that are firstly the destination of question addressed by user, secondly the provider of the answer to user question and finally information sources to extract answers(cf. Figure 2.1).

The destination of question addressed by user, that is whether user supposes human or supposes computer as a party of question-answering, conditions the input specification of question-answering systems. Current computer systems have not achieved the same level of intelligence as humans, users of question-answering system have to realize the forms of questions such that the question-answering system can accept. Looking at the same fact from question-answering system, it means that question-answering system does not necessarily identify the question such that a human describes on the assumption that other human reads and answers the question. For instance, if most of questions are stereotyped,

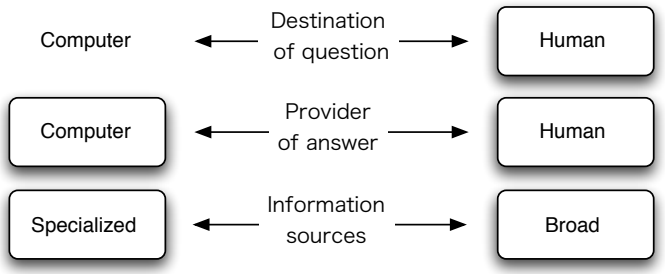

Figure 2.1. Three Viewpoints to Environment of Question-Answering.

keyword inputs in traditional information retrieval and input forms of search condition in database retrieval are possible forms of queries.

The provider of answer to user's question restricts task performed by question-answering system. For instance, if humans always provide answers to a question, the system only supports the humans who provide answers.

The information sources to extract answer conditions the types of question and the range of application. If a question-answering system aims to answer questions related to certain domains and topics, required information sources are just that comprising contents related to the domains and topics. Contrarily, if domains and topics of question are unrestricted, it is necessary to equip information source comprising more broad contents. For instance, the World Wide Web is a typical information source containing broad contents. Additionally, the degree of difficulty to find answers is different regarding types of information sources. For instance, Q&A document contains relatively refined articles and directly exploits the association betweens questions and corresponding answers. In case of web pages, such prosperous conditions are however not existed.

The question-answering system in this thesis supposes conditions of the environment as follows,

- The question-answering system accepts questions such that a human describes on the assumption that other human reads and answers the question.
- A human does not only provide the answer but a question-answering system also answers to the same question.

- Domains and topics of question are unrestricted
- Answers are extracted from Q&A documents and Web documents

For question-answering systems in the environment satisfying those conditions, two applications are targeted in this thesis; automatic question-answering in e-mail-based help service and push-based question-answering service. In recent years, along with centralization of customer support in online, management of large amount of e-mail and comments in the Web site of help service from customers became serious problems [70, 178]. In this kind of service, operators swamped with large amount of questions or requests, thereby question-answering systems are required to provide answers to a part of questions customers address to operators as much as possible. Push-based question-answering service works under an environment of question-answering in that answers are automatically linked to any questions without designating certain answerers. In this environment, question-answering services always check user articles in blogs and diaries in Web pages and in posted e-mails, and then regularly extract question parts from their articles. Proposed question-answering system in this thesis extracts question segments, identifies their question types, and then extracts the answers from information sources. The question-answering service links extracted answers to the questions in blogs and e-mails. The information of the Web often is problematic in the credibility. Moreover, along with increase of this kind of push-based question-answering services, the spam links are likely to increase. Therefore, I consider the environment of question-answering in that other humans also provide answers to the same questions. Additionally, it is necessary to establish the framework for evaluation of answers and to control links between questions and answers in any secure online community [115, 134].

The question-answering systems in those kinds of environments have to handle questions such that a human describes on the assumption that other human reads and answers the question. Additionally, to avoid restriction of domains and topics of question, question-answering system needs to exploit information source comprising broad contents such as the Web in addition to specialized information sources; Q&A documents [15, 79, 146] and databases such as patents [122] and legal information [106]. This thesis aims at a question-answering system

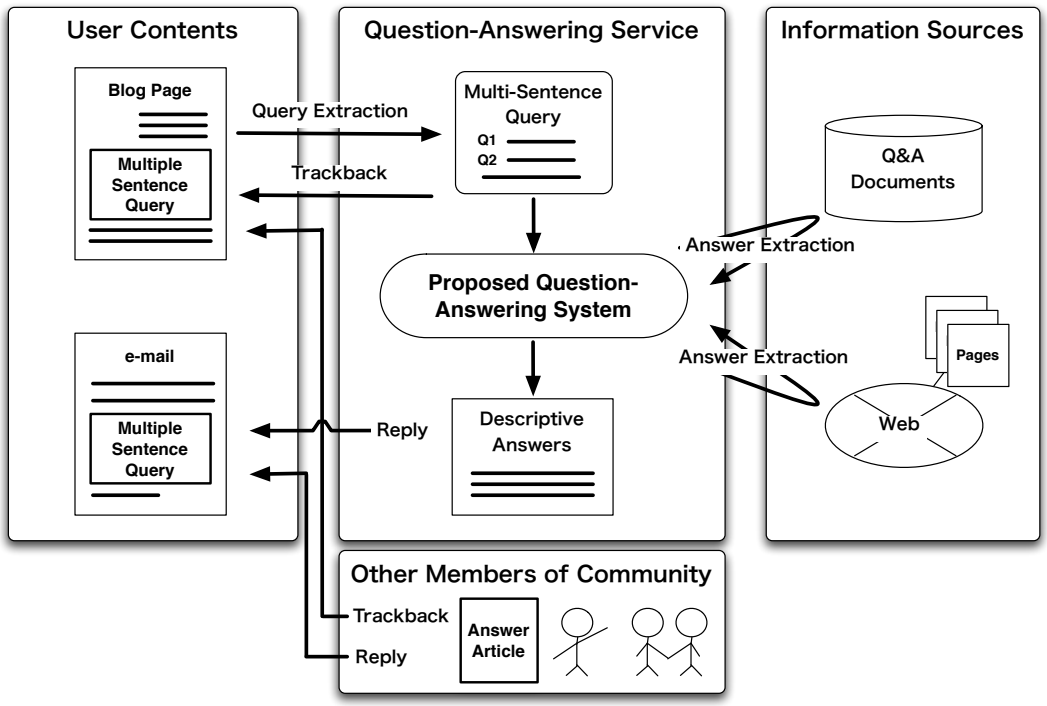

Figure 2.2. Push-based Question-Answering Environment.

that effectively works in this kind of environments. On the other hand, question-answering system aimed in this thesis does not suppose specific user interfaces, that are multi-modal interfaces such as image presentation and speech recognition [47, 48, 111, 112]. and mobile devices such as cell phones [3, 4, 19, 99, 164, 182]. The question-answering system in this thesis is regarded as a functional unit in a larger system such as e-mail-based automatic question-answering system and push-based question answering service. Therefore this thesis pays not much attention to the primary query form by user and final answer form provided to user. The target question-answering system in this thesis always expects narrative text comprising one or more sentences as a query and cases in that question type identification can be performed using only the input query. If any optional information to identify question type is required, any external components of question-answering system such as dialogue system complete insufficient information. As well as input queries, question-answering system in this thesis does not perform

editing or personalizing the outputs according to attributes of output devices and user. If necessary, external components summarize multiple answer texts or single answer text [8, 11, 36, 44, 81, 83, 89–92, 96, 107, 108, 119, 139, 160, 191] and generates answer texts [114, 171]

To handle queries requiring descriptive answer, the question-answering system claims the architecture described in Figure 2.3. This system consists of process of question analysis introduced in Chapter 3, the identification of description type described in Chapter 4, the extraction of descriptive answer from huge amount of documents examined in Chapter 5, and rest of processes summarized in Chapter 2. Besides the processes discussed in Chapter 3 to Chapter 5, such as matching the question and answer are constructed by conventional techniques described in Chapter 2.

2.2 System architecture of target question-answering

In Figure 2.3, the process flows the left to the right. It is also divided lengthwise to three areas; the upper area shows input data, and the middle contains process flow, and the lower area presents features and resources to each process. The five phases of processes are horizontally aligned and the ellipses in the center of phases represent main components conducting the processes.

Question type identification
First of all, when a query is given to the question-answering system, a chunker in question segmentation phase decomposes the query into questions and then identifies their question types. Additionally, this phase yields keywords and patterns exploited in matching questions and answers, and stores them with the sentences in the question segment list of the query. When the chunker fails to identify the question type or to extract any question, the system activates counter processors that attempt to identify the question type regarding the query as one question. The question patterns are mainly exploited in pre-processing of question type identification to divide a sentence comprising multiple questions. In Chapter 3, this process will be discussed in detail .

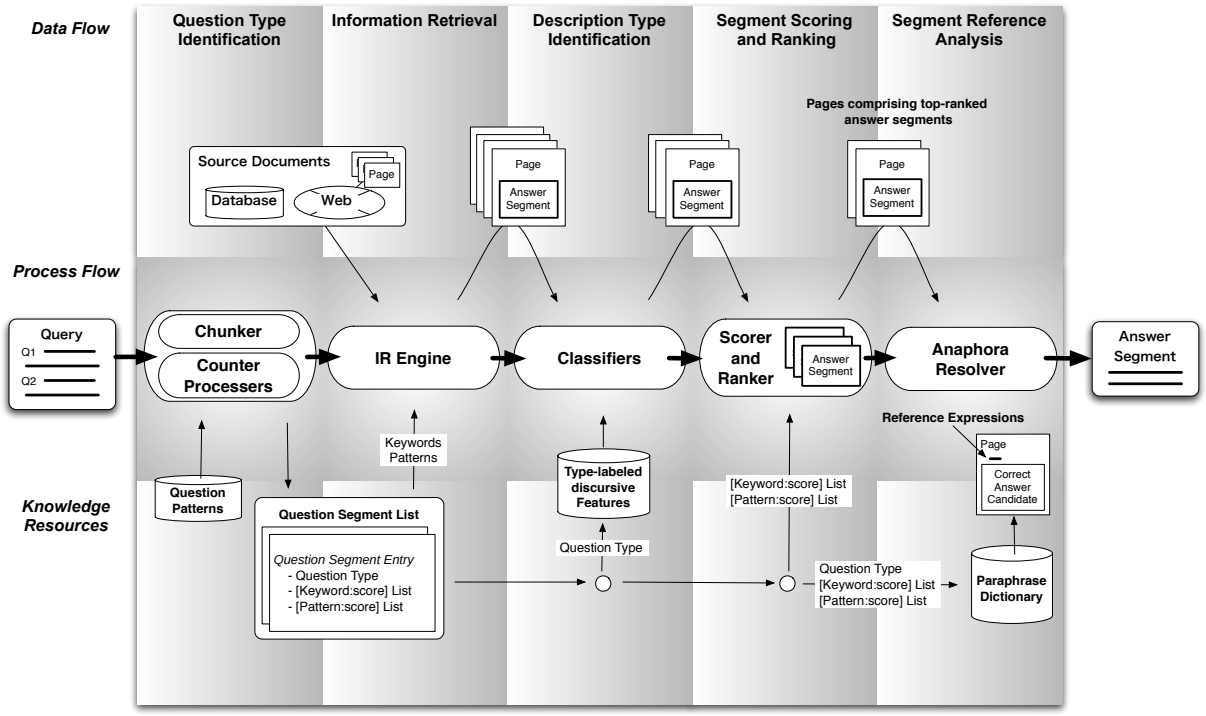

Figure 2.3. The System Architecture of Target Question-Answering System.

Information retrieval

In next phase, the system retrieves source documents comprising answer candidates with keywords, patterns and question types in question segment list of questions. Information sources are both web pages and database entries such as pairs of question and answer in Q&A documents.

Description type identification

After the information retrieval phase, the system classifies answer candidates based on their description types related to their question types. To perform this process, it exploits discursive features of text that specifies description types, such as cue words, patterns of functional words and so forth. Consequently, the system narrows down their candidates into more relevant set of candidates. This phase will be more clarified in Chapter 4.

		Core Techniques							Web				AI		Ontology			
Fundamental techniques	IR and NLP Technologies	Passage Retrieval	Text Segmentation	Text Categorization	Anaphora Resolution	Noun Phrase Analysis	Text Summarization	Sentence Alignment	Text Cleaning	Page Segmentation	Wrapper Induction	List and Table Detection	Semantic Tagging	Organization	Pattern Acquisition	Lexical Knowledge Acquisition	Corpus Anotation	Machine Learning
Phase of Process	Question Type Identification		○	○	○	○			○	○	○				○	○	○	○
	Information Retrieval	○		○	○	○			○	○	○	○	○		○	○	○	○
	Description Type Identification			○			○		○	○	○	○	○		○		○	○
	Segment Scoring and Ranking	○	○	○		○	○	○						○		○	○	○
	Segment Reference Analysis				○	○									○	○	○	○

Figure 2.4. System Components and the Related Work.

Segment Scoring and Ranking

Moreover, the system computes the relevance of answer candidates to the question using keywords and patterns acquired in the first phase. Subsequently, the question-answering system ranks the candidates and selects top-ranked answers.

Segment Reference Analysis

Finally, the system justifies the answer candidates by checking expressions referring to them, and outputs the relevant answer. The variety of reference expressions is solved using a paraphrase dictionary of reference expressions.

Figure 2.4 shows the relation between internal components in Figure 2.3 and fundamental technologies in the following Sections. In the rest of this chapter,

I will introduce fundamental technologies of information retrieval and natural language processing.

2.3 Fundamental technologies and related work

When seeking answers in an information source, a QA system does not always exploit the whole articles. Typically, segments that are expected to include the answer are extracted, while eliminating the parts composed of noise and unnecessary information form the sources. These technologies are called text segmentation or page segmentation [17, 43, 100, 161, 179] or passage retrieval [61]. If the source contains much noise, as seen in Web documents, text cleaning [148] should be performed prior to segmentation.

Although the segmentation scheme is diverse, so-called tables and lists are useful to find answers. Because they are kinds of summarization of information sources, it can be expected that they contain answers. Several techniques for finding tables and lists in a document, table and list detection [4, 93, 164, 165] has been proposed [37, 79, 102, 103]. After segmentation, the text clustering or text categorization is performed to classify segments by the topic and domain [132, 145, 186]. When necessary, sentence extraction [133] is conducted.

The processes mentioned above are often invoked by diverse heuristic rules [41, 94]. There exist approaches, such as wrapper induction, that automatically or semi-automatically acquire such kind of rules from the documents [21, 142, 167].

The process of extracting questions and answers from a sentence heavily incorporates various techniques and resources of natural language processing. Sentence type identification [63, 138, 147] and anaphora resolution [35, 39, 52, 53, 64, 65, 98] are often conducted. To extract phrases that could be candidate answers, especially in the case of a factoid question, named entity recognition [104, 118, 123, 127] or noun phrase analysis [1, 159, 176, 188] would be performed. In this processing, a large electronic thesaurus and dictionaries [27, 54], chunkers [74, 144] and some kind of parsers [76] may be incorporated. Moreover, many kind of mining technologies that acquire patterns are used to extract answers from source articles by pattern matching, and obtain knowledge for named entity recognition [62, 74, 75].

After extraction of answer passages, based on the result of identification of relation and similarity between segments, sentences and passages, alignment and organization are invoked [40, 55, 101, 109, 121, 131, 174].

Stochastic machine learning is heavily used as an underlying methodology to execute the natural language processing shown above [20, 22, 23, 25, 29, 67, 78, 84, 153].

Recent natural language processing, however, is not yet at such a level analyze the meaning represented by natural language, so there are still a number of problem hardly solved with natural language processing alone. In addition to natural language, if we could exploit additional information presenting the meaning, such as semantic annotation, the accuracy and coverage of question-answering would be improved [42]. In the Web information retrieval, the development of a tagging scheme based on a semantic web [12] has proceeded [86].

In the QA system dealing with queries that require descriptive answers, many kinds of tagging scheme have been used for acquiring linguistic knowledge exploited in question analysis, answer extraction, summarization of answers, and so on [10, 16, 45, 46, 56, 57, 85, 110, 140, 143], because linguistic knowledge to identify logical or rhetorical relations between sentences are necessary. Lately, annotation schemes for spoken language have caught attention of many researchers [49, 180].

The design of the annotation scheme should be discussed along with annotation tools and the environment. There have been many studies looking at the efficiency of making a corpus and sharing knowledge for relevant annotation between annotators [50, 97, 173]. Additionally, there is a problem of how to manage annotation results such as disagreement of annotations between annotators [105, 129].

Chapter 3

Question Type Identification for Multi-Sentence Queries

3.1 Introduction

Question type identification is a question analysis executed by question answering systems, information retrieval, dialogue system, and other applications. Question analysis provides a variety of information from query inputs into the application system, and converts the queries into their formats required for the internal processing of the application. Question type identification is a process to extract questions from a query given in a natural language sentence and identifying its intention with other operations in question analysis. This processing is the initial stage of the internal processing flow of the application, thus its accuracy exerts a major effect on the accuracy of the entire application. This Chapter describes question type identification in question-answering, but it is applicable to other applications requiring question type identification such as information retrieval.

Diverse question types handled by question type identification are proposed in conjunction with the queries permitted as input by applications. In the field of information retrieval, the Text Retrieval Conference (TREC), an international evaluation campaign, and the NII Test Collection for IR Systems (NTCIR)[1] have been researching question-answering for large-volume documents in any field. It

[1] http://research.nii.ac.jp/ntcir/index–en.html

is called open domain question-answering as new information access technology. In their researches, question- answering is a technology capable of extracting an appropriate answer from large-volume documents to a question given in a natural language sentence and generating the answer meeting the purpose. Queries processed in TREC QA Track and NTCIR QAC are input with single-sentence questions, and their question types are semantic category, such as personal names and place names that their answers belong to. Answers are mainly those given by proper noun phrases or single sentence.

On the other hand, in the actual fields requiring question type identification, such as call centers of enterprises and internet information services, frequently handle multi-sentence queries. Additionaly, single query often includes multiple questions. The question types handled are not only those provided answers by noun phrases or sentences but also those answered by multiple sentences, such as methods and opinions.

If a multi-sentence query includes multiple questions, each question must be extracted from the query in order to identify the question type. Question type identification handling such kind of queries differs from the question type identification handling just question sentences in some important aspects.

In a multi-sentence query, the information required to understand the question is often divided into multiple sentences. On the other hand, multi-sentence query contains contents that are not directly used for question type identification, such as greetings or apologies. For extracting only sentences required for question type identification, irrelevant sentences for question type identification have to be removed so that the question type can be correctly identified. With regard to a query including multiple questions, the relations between them are also important. If they are relative to one another and their relations are correctly identified, it can be used for selection of a answer.

Although some previous researches have been conducted into the question type identification of multi-sentence queries, many of them rely on pattern matching. Open domain QA must handle a variety of questions, thus approaches requiring patterns to be manually created are costly.

This Chapter presents an approach to question type identification as a chunking problem of sentences, which combines N-gram of words and other features

used for question sentence type identification by a learning-based approach with conditional random field (CRF).

I performed evaluations and experiments, and investigated the effectiveness of the proposed approach. We also report herein the accuracy of sentence extraction required for question type identification and the accuracy of question type identification separately, as well as the results of analyses of individual effective features.

3.2 Question segmentation and type identification

Figure 3.1 shows an example of a multi-sentence query. In this example, the numbers given on the left of the sentences are the sentence numbers assigned from the head of the query. The single query includes two questions, one described by sentence (2) and one by sentences (5) and (6). In this Chapter, a set of sentences describing a single question such as (5) and (6) is called a question segment. Therefore, the query shown in Figure 3.1 includes two question segments. A variety of question segments is conceivable: however, in this chapter, it is assumed that a question segment is the shortest series of sentences describing a question. Question type identification herein means extracting question segments and identifying their question types.

In international evaluation campaigns such as MUC(Message Understanding Conference[2]), TREC, and NTCIR, diverse question types have been proposed for a question sentence. The question types concerning multi-sentence queries are defined for distinguishing question sentences from other sentences [60], or based on question focus such as 5W1H [178], aforementioned two types [70, 77], or question types including questions that require descriptive answers [141, 185] and so forth. Since this Chapter is intended to cover questions that require a descriptive answer, we set ten unique question types based on the question types proposed by Tamura, and others [141, 185]. Table 3.1 shows the definitions of the set question types.

[2]http://www–nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Table 3.1. Definitions of Question Types.

Question Types	Definitions	Examples
Yes-No	A question demanding an answer of yes or no	Is there a SAVE-button in the browser?
Name	Asking a name except of a place	Who was the US's first president?
Description	Asking a definition, attribute, property, aspect, number, amount and degree	In the case of infection with hepatitis, what kind of symptoms appear?
Evalauation	Asking an opinion	How is the digital camera of company A?
How-to	Asking a method	What should I do when I want to install Internet Explorer?
Reason	Asking a reason	Why is an OS necessary?
Location	Asking a place	Where is Canada's capital?
Time	Asking a time or period	When was the Nobel-price established?
Consultation	Question that matches several of the above types in a same time	Can you take a longer holiday this summer? Don't you know anything fun?
Other	Question not falling under any of the above types	

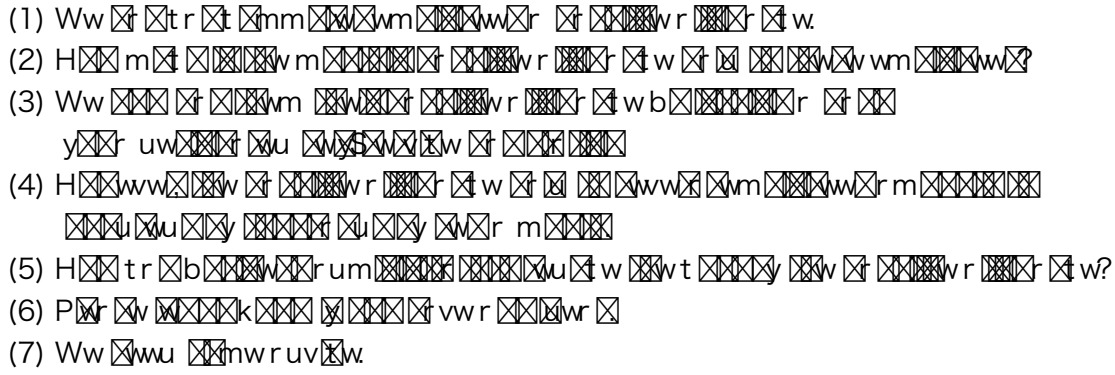

Figure 3.1. Example of a Multi-Sentence Query.

It is not clear what characteristics are effective in extracting a question segment from a multi-sentence query and identifying its question type. The characteristics for question type identification in the previous researches must be reviewed in an evaluation of question segments including multiple sentences. We, therefore, annotated actual multi-sentence queries and reviewed what characteristics were necessary for question segment extraction and question type identification.

3.3 Question type annotation to multi-sentence queries

3.3.1 Overview of corpus

Today, question answering services that provide answers for questions from unrestricted users, such as “Yahoo! Chiebukuro” “Hatena” “Oshiete! goo” are available on the Internet. We studied 2,234 queries obtained from the question answering website “Oshiete!goo.” This website offers online services through which anybody can ask or answer questions using forms uploaded on its home-

[1]http://chiebukuro.yahoo.co.jp/
[2]http://www.hatena.ne.jp/
[3]http://oshiete.goo.ne.jp/

page. Past questions and the answers are categorized into Life, Hobby, and so forth.

I selected 21 categories, and collected 200 latest queries as of July 21, 2006 from each category, consequently gathered 4,200 queries in total. The 21 categories include gardening, town/local information, healthcare, law, economy, mass media/communication, news, social issues, politics, history, archeology, Japanese language, biology, automobiles, domestic travel, stocks, restaurants/eating houses, software/freeware, finance/accounting, side jobs/part-time jobs, and mental health. From the obtained queries, I selected 3,993 queries to which answers were given and subsequently chose queries including at least two sentences. After confirming the contents and excluding the queries that questions were indefinite, consequently, I obtained 3,628 queries. We further sampled 2,000 queries of the 3,628 queries at random, and created sets of queries for annotation. Besides the 2,000 queries, we used 234 queries that were collected in 2001 for research from six categories (gardening, healthcare, economy, sociology, politics, and law) on the same website in the same manner. The data sets thus created are 5.7 in the average number of sentences per query and 3.9 in deviation. The average length and deviation of a sentence are 73.9 bytes and 51.8 respectively.

3.3.2 Overview of question type annotation

Question types were manually tagged in the ten kinds of question types listed in Table 3.1. The annotators tagged passages considered as necessary to identify one question and its question type. Consequently, one question was expressed by a set of several text passages. The boundary of tagged passages were allowed to be at any character and not necessarily located to be at the start or end of a sentence. It was allowed to only assign one question type to one passage. For this reason, nonoverlapped passages tagged in different question types could be contained in one sentence. The query was presented to the rater without showing its answer or question title.

Tagrin[3] was used as the question type annotation tool [173]. The corpus was divided into two and the respective articles were classified by two operators. Furthermore, 234 queries collected in 2001 were tagged by another operator besides

[3]http://kagonma.org/tagrin/docs/main.html

Table 3.2. Classified Given Question Types.

Question-Types	Number of Passages
Yes-No(Y)	1709 / .43
Description(D)	636 / .59
Name(N)	454 / .71
How-to(W)	325 / .79
Reason(R)	304 / .87
Location(L)	197 / .92
Evaluation(E)	141 / .95
Consultation(C)	106 / .98
Time(T)	63 / 1.00
Oters(OT)	10 / 1.00
Total	3945

above-mentioned two operators. The question type annotation results were compared with those of the other two persons to calculate inter-annotator agreement.

3.3.3 Analysis of assigned tags

The results of question type annotation according to the settings described in Section 3.3.2 are shown in Table 3.2. The right column in the table indicates the frequency of tagged passages for each question type where they are arranged in the descending order of frequency from the top. The adjacent values of each frequencies indicate their cumulative ratio of frequencies to the total frequency of all passages.

In total, 3945 passages related to questions and 1252 articles each containing more than one question item were confirmed, and the number of question items per article was 1.77. There were 98 questions where the passage corresponding to one question item was contained in more than one sentence. There were 188 sentences each containing more than one question item, accounting for about 5% of all sentences containing question items.

3.3.4 Combination of question types in a query.

Table 3.3. Definition of Classes of Question Types.

Yes-No	Yes-No
Factoid	Name, Location, Time
Mixed	Description, Consultation
Non-Factoid	How-to, Evaluation, Reason

Figure 3.2 shows a ratio of frequencies in the combinations of question types within a query. The labels in this bar chart, that are Yes-No, Factoid, non-factoid and Mixed, are defined in Table 3.3.4, which indicate classes of annotated question types. Mixed types are defined for Description and Consultation question types, because these two types can be classified to both factoid and non-factoid questions depending on the contents of question. These classes of question types are assigned to horizontal and vertical axes in the graph, each bar indicates a ratio of co-occurrence frequency corresponding to question types in both axes to the frequency of the question types in vertical axis.

As shown in Figure 3.2, the cases that the same question types co-occur in a query appear the most frequently in all classes of question types. The chart also indicates that Yes-No type frequently occurs compared with other classes, i.e., Factoid, Non-factoid and Mixed. Especially, Yes-No type occurs more frequently in Mixed than in Factoid and Non-factoid. Contrarily, in the case of queries comprising Yes-No type questions, other three types occur in about similar frequencies of co-occurrence. Besides Yes-No types, there is no salient difference of ratio of co-occurrence between two different question types of Factoid, Non-factoid and Mixed.

3.3.5 Inter-annotator agreement for question type annotation

The agreement for question type annotation was calculated sentence-by-sentence. Question type was annotated for passages, consequently, the question type for a sentence is not confirmed in this state. The question type of a passage is assigned

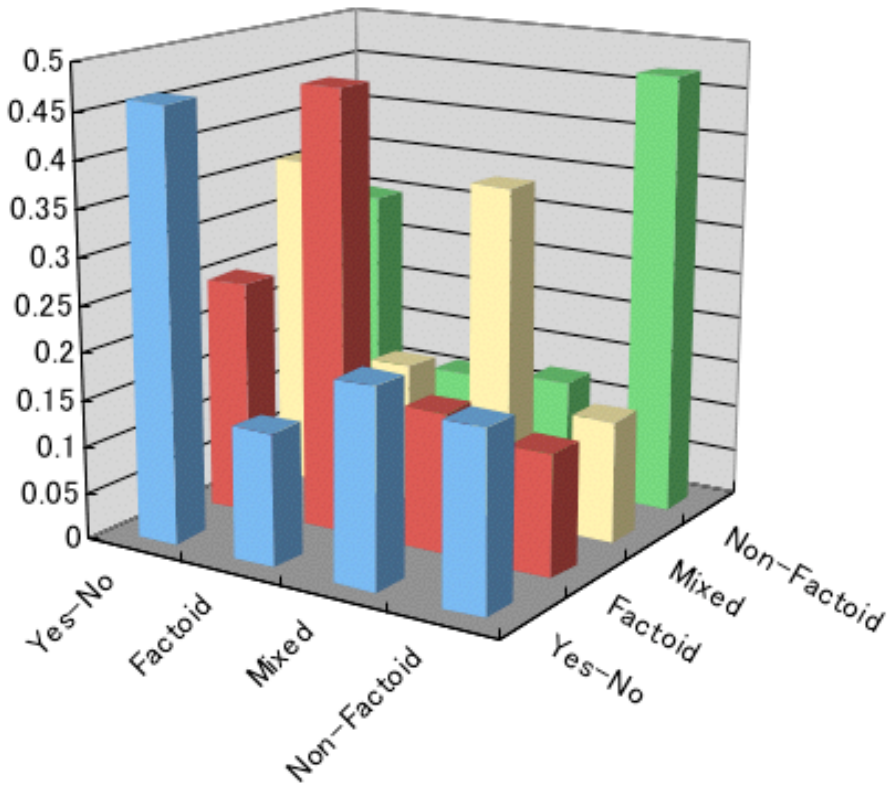

Figure 3.2. Combinations of Question Types in a Query.

to the sentence that contains the passage. A sentence containing more than one question item was handled as having more than one question type. In this case, the agreement for question type annotation was assumed to agree when all the question types of the sentence matched. The F-measure[4] as used in the evaluation of MUC was used for the inter-annotator agreement for question type annotation. The F-measure can be defined by equation 3.1.

$P(t)$ and $R(t)$ are calculated according to the equations 3.2 and 3.3 where the numbers of questions, which annotator A and annotator B classified into question type t, are represented by $C(A,t)$ and $C(B,t)$, respectively, and the number of questions, which both rater A and rater B classified into question type t, is represented by $C(A,B,t)$.

[4]http://www–nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/muc7_score_intro.pdf

$$F(t) = \frac{(\beta^2 + 1.0) \cdot P(t) \cdot R(t)}{(\beta^2 \cdot P(t)) + R(t)} \tag{3.1}$$

$$P(t) = \frac{C(A, B, t)}{C(B, t)} \tag{3.2}$$

$$R(t) = \frac{C(A, B, t)}{C(A, t)} \tag{3.3}$$

β is the coefficient that determines the relative weight of $P(t)$ and $R(t)$. In this test, β was set to 1, giving the same weight to $P(t)$ and $R(t)$.

After calculating the inter-annotator agreement for question types, it was found that there were variations in concordance rate depending on the question types. The F-measure was 0.7 in the Yes–No type and the Location type with the highest concordance rate, and 0.5 in the Name type and the Description type with the lowest rate. The Evaluation type, the Time type, and the Other type could not be evaluated because of the small number of case examples. For sentences containing more than one question item, all the tagged question types need to match and therefore, the agreement tends to be low. When the agreement was calculated excluding the sentences containing more than one question item, the F-measure was 0.8 in the Yes–No type, the Location type, and the How–to type with the highest agreement, and 0.5 in the Description type with the lowest agreement . Besides, a relatively high agreement was obtained in the Reason type.

3.3.6 Question extending beyond more than one sentence

Although questions consisting of more than one sentence were observed, such questions were few relative to the entire number of question items. In some articles, there are question items in that the interrogative sentence is different from the sentence describing the contents of the question, as shown in Figure 3.1 in Section 3.2. In such a case, the tags were given to both sentences. Figure 3.3 shows a case where restrictions concerning the question are added one-by-one using another sentence and an answer is required to meet both conditions of the first sentence and second sentence. In such a case, only the second sentence cannot cover the conditions as requested by users.

＜名称＞室内でも育つ植物。加えて、初心者でも失敗の少ない植物
教えて下さい。＜/名称＞

<name> Plants can grow indoors. In addition, plants could you introduce to me < /name> which even beginners are not likely to fail to grow?

Figure 3.3. Example of Question Crossing over Multiple Sentences.

3.3.7 Sentence containing more than one question item

One hundred questions were selected at random from sentences each containing more than one question item and their linguistic characteristics were classified. Of those questions, more than 80% corresponded to one of the five types as follows;

Type 1 Coordinate clauses marked by conjunctions.

Type 2 Continuous clauses patternized in sequences of nominals including interrogatives, another nominal and particles.

Type 3 Continuous clauses patternized in sequences of interrogatives, particles, and punctuations.

Type 4 Continuous clauses after nominals including interrogatives.

Type 5 Noun phrases connected by particles in parallel.

Table 3.4 shows examples of relations between questions in a sentence. In the example of Type 1, the second question was asked based on the assumption that the first question was answered. In this type, the correct answer cannot be selected only when individual questions are simply extracted one-by-one to individually identify the question type.

However, besides the Type1, in their expression such as in Table 3.4, certain patterns can be recognized. Thus we assume that processing of sentences containing more than one question item can be handled by relatively simple pattern processing. Although identification of the relations between multiple related questions are necessary in the case of Type 1, such cases are not discussed in

Table 3.4. Examples of Relation between Questions in a Sentence.

Type	Question Segments
1	<HOW-TO>化石を破壊することなく年代測定する方法はあるのでしょうか</HOW-TO>、あるのでしたら<DESCRIPTION>どの程度正確に測定できるものか教えていただけませんか？</DESCRIPTION> <HOW-TO>Are there any non-destructive dating techniques for fossils?</HOW-TO> <DESCRIPTION>If exists, how precisely does it date them?</DESCRIPTION>
2	その男性は、<TIME>いつの人で</TIME>、<NAME>何をしていた人でしょうか？</NAME> <TIME>What era</TIME> did he live in and <NAME>what did he do in that time?</NAME>
3	近所のお祭りに行きたいのですが、<TIME>いつ</TIME>、<LOCATION>どこでお祭りがやっているのか教えていただきたい</LOCATION>のですが…。 I would like to go a festival near here. Do anyone know <TIME>when</TIME> and <LOCATION>where</LOCATION> any exciting festivals hold?
4	その頃からこの言葉が使われ始めた思うのですが、<TIME>何年の</TIME><DESCRIPTION>どのような場面から使われだしたか</DESCRIPTION>教えてください。 Since that time, people may used the word. <TIME>What year did it begin</TIME> and <DESCRIPTION>what trigger did happen in the time?</DESCRIPTION>
5	<NAME>オススメの問題集</NAME>や<HOW-TO>勉強法</HOW-TO>などありましたら、ぜひ教えて下さい！ Please send me any information about <NAME>exercise books</NAME> and <HOW-TO>learning methods</HOW-TO>

this thesis. This Chapter proposes a simple model for identifying the question segment and question type as a study aid in identifying the type of multi-sentence queries.

3.4 Chunking-based question segmentation and type identification

The question type annotation conducted in Section 3.3 was a task in which humans extracted question segments and identified question types at the same time. However, these two processes do not necessarily have to be performed at the same time. For type identification of multi-sentence queries, Tamura et.al.[141] have proposed a two-step method by which the question segment and question type are separately identified. Their study is the only previous study concerning the matters discussed in their paper as far as we know. With the method by Tamura et.al., the question segment is limited to one sentence and no identification of question type for the cases containing more than one question has not been proposed. Their method is a learning-based method using SVMs(Support Vector Machines [153]) and the features effective in question segment extraction and question type identification have been analyzed in detail. Therefore, their report is worth consideration in this thesis. According to their report, the experimental results obtained when segment extraction (called core sentence extraction) and question type identification are separately performed are better than those obtained when they are performed at the same time. If this is true, there should be a condition that is effective in question segment extraction but disadvantageous in question type identification, or vice versa. In the Tamura's tests, better results were obtained by using not only the features of the sentence to be extracted but also the preceding and following contexts. In addition, it was reported that the longer the context to be used the better the accuracy became. If this is correct, better results may be obtained by extracting a question segment considering not only the local context around the question segment subject to extraction and identification but also the context of the entire question article.

However, the study by Tamura et.al. remains problematic. With the SVM-based method proposed in their paper, it is predictable that there will be an

considerable difference in computational cost between the cases when question segment extraction and question type identification are performed at the same time and when not. In the method expanded to the case of query containing more than one question item, the sign and score that the SVM gives to the core sentence are used as a criteria for selecting a core sentence containing question items [185]. However, no evidence has been presented to prove that core sentences can be properly extracted according to such a criteria. Moreover, an assumption that a question segment consists of one sentence is a more fundamental problem that is different from observations of real queries.

In an attempt to solve these problems, we propose a method by which question segment extraction and its question type identification are performed at the same time to solve the sentence-chunking problem using Conditional Random Fields (CRF, a machine learning method). This method is capable of executing question segment extraction and question type identification at the same time and is also advantageous in terms of computational cost. This can also apply, in a natural way, to query articles containing more than one question segment. As compared with SVM, CRF has the property of selecting the optimum model for the entire space of solutions and therefore, it should be advantageous in the tasks discussed in this thesis. It has been reported that CRF has higher performance in several tasks than SVM and therefore, CRF is comparable to SVM, as a learning algorithm. This section describes question type identification based on sentence chunking using CRF.

3.4.1 Chunking

Chunking is a process of identifying chunks that indicates some sort of visual or semantic unit. Chunks as used in the field of natural language processing often indicate the noun phrase and paragraph, or lexical and grammatical units. In this case, chunking is a processing which forms morphemes and sentences into chunks such as noun phrases and paragraphs.

Although there are various expression of expressing chunks, we adopted the method by which a tag indicating the status of a chunk is given to each sentence, which permits modeling in the same framework as for the conventional problem of tagging to morphemes and noun phrases. For this task, previous methods such

as Inside/Outside [113, 116] and Start/End [151] have been proposed. Kudo et. al.[73] summarized them into five expressions of IOB1, IOB2, IOE1, IOE2, and IOBES(Start/End). First, the following ten kinds of chunk statuses are defined.

I1 The word in the present position is part of the chunk.

I2 The word in the present position is a middle word other than at the start or end of the chunk consisting of three words or more.

B1 The word in the present position is the start of the chunk immediately following a chunk.

B2 A tag is given to the start of every chunk.

B3 The word in the present position is the word at the start of the chunk consisting of two words or more.

E1 A tag is given to the word at the end of the chunk immediately preceding a chunk.

E2 A tag is given to the word at the end of every chunk.

E3 The word in the present position is the word at the end of the chunk consisting of two words or more.

S The word in the present position singularly consisting of one chunk.

O The word in the present position is not included in the chunk.

At this time, IOB1, IOB2, IOE1, IOE2, and IOBES are models that perform tagging to meet the combinations of the following rules based on the above rules.

IOB1 I1, O, B1

IOB2 I1, O, B2

IOE1 I1, O, E1

IOE2 I1, O, E2

IOBES I2, O, B3, E3, S

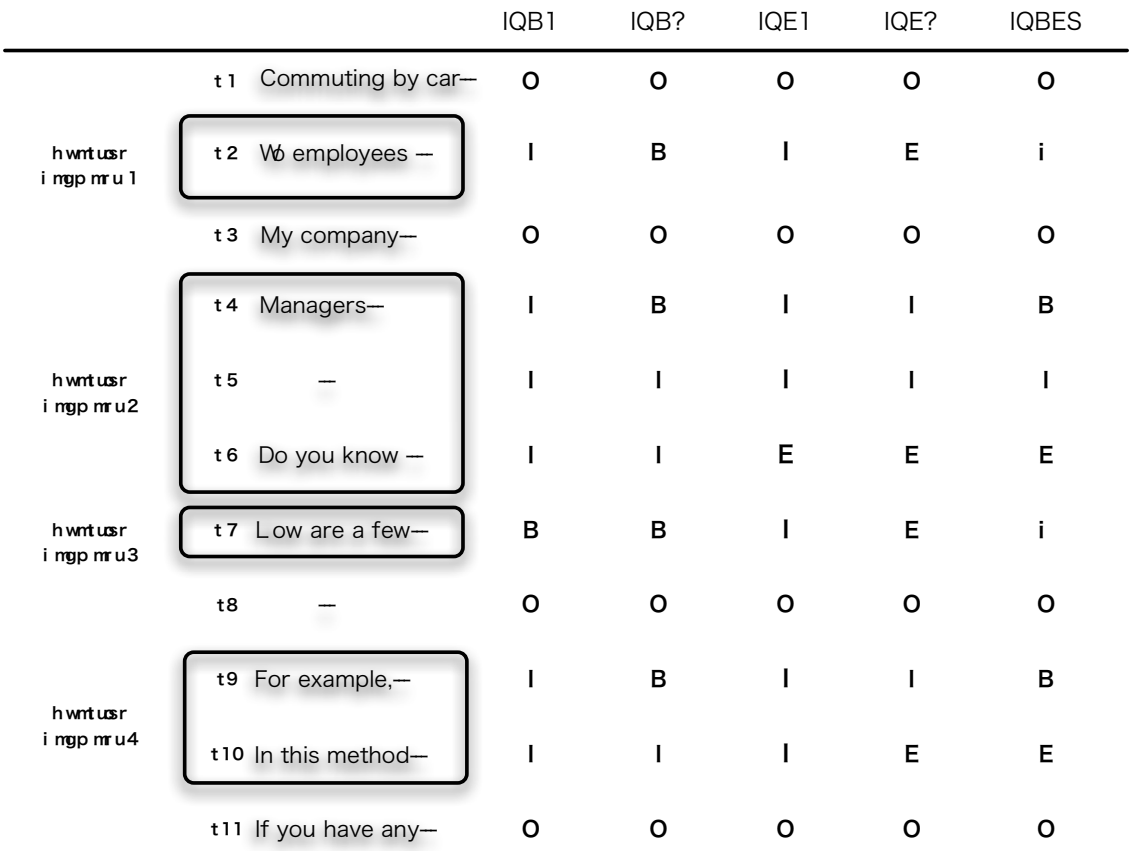

Figure 3.4. Example of Assignment of Chunk Labels.

Actual tagging by IOB1, IOB2, IOE1, IOE2, and IOBES are shown in Figure 3.4.

In order to indicate the question type of chunk, a tag indicating the question type is linked to a tag indicating portion in the chunk such as B, E, I, O and S with a hyphen "-". For example, the B-W of IOB2 in Figure 3.4 is given at the start sentence of question segment 4.

3.4.2 Overview of the proposed technique

The processing flow in the proposed technique of chunking follows the steps in the list below.

Step 1 Segment a question article into sentences. Each segment is terminated with a period ".".

Step 2 Carry out chunking by article.

Step 3 Extract question segments as chunks, identify the question types, and output them in pairs.

Chunker divides a sequence of sentences into question segments and other chunks. A chunk tag is given to each sentence. The chunk tags used are of the five types explained in Section 3.4.1, namely IOB1, IOB2, IOE1, IOE2, and IOBES, and the IO-tag that does not distinguish the B/E/S tags from the others. Sentences not involved in the identification of question types are given the O-tag. Those sentences that constitute a question segment are given a tag consisting of the combination of one of the letters I, B, E, and S and one of the letters W and D, thus I-W and B-D for example, to represent the portion in the chunk and the question type. Figure 3.5 shows an example of composition of chunks using the IOB-tags. A chunker learns a chunking model from the pairs of sentences and their chunk tags in Figure 3.5. To extract question segments from a query, sentence labeling, that labels a chunk tag to a sentence, is firstly performed. Subsequently, sentences labeled same roles such as "-D" and "-W" are chunked by post-processing. Consequently, a question segment is extracted as a chunk and the question type is given to the question segment based on the label of chunk.

3.4.3 Conditional random field

The CRF (Conditional Random Field) is a stochastic model for sets. Combinations of two random variables to represent the properties of a set are associated with each other as a conditional probability [Lafferty 2001]. The CRF supposes a random field that has the Markov property regarding the elements of a set to be observed. The advantages of this are as follows: (1) There is no need to assume the independency of random variables as with those in the Markov model; (2) Since a model is described with conditional random variables, the model parameters can be estimated without calculating the distribution of random variables in the condition.

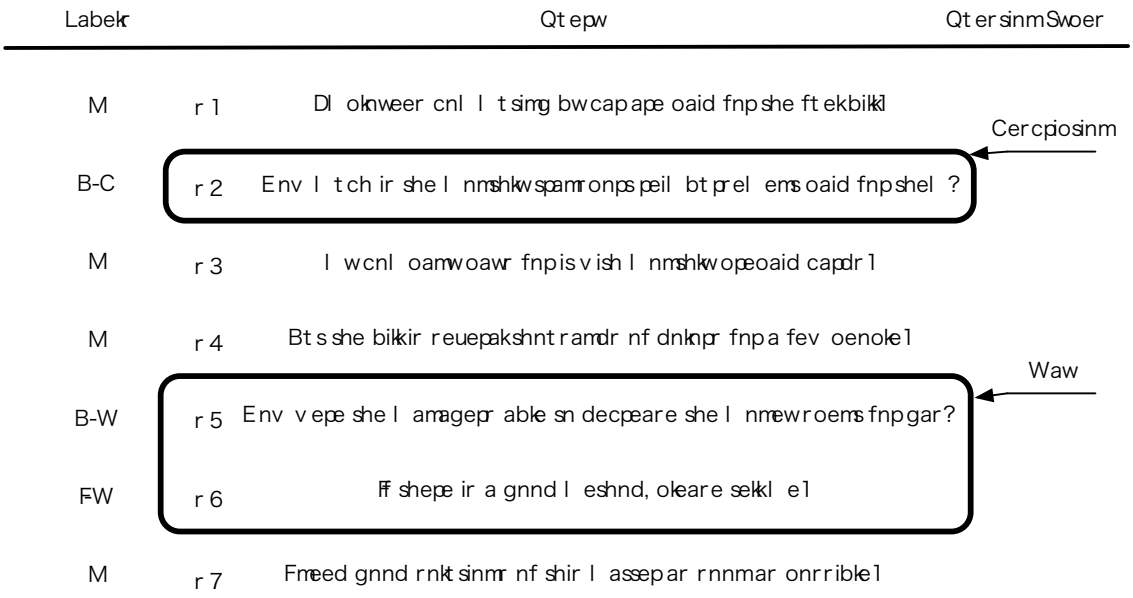

Figure 3.5. Extracting Question Segments and Identifying Question Types.

One report points out that the CRF provides a performance similar to that of the HMM with a number of cases less than that needed for the HMM in the order of sample of 1 to one-several-tenths [172]. Taking advantage of these advances, the CRF has been used in natural language processing and bioinformatics. The CRF, however, is unable to forecast from an estimated model those input variables that are set as conditions, it cannot be applied to a case that requires regeneration of instances based on a model.

The model of CRF is described with a feature function defined by two random variables: one to represent conditions and the other to represent a random field. The following paragraphs explain the CRF based on explanations by Kashima et.al.[172], which takes for example a case where the CRF was applied to a labeling to sequential symbols.

For a set of feature function F, let the number of locations where feature $f \in F$ holds for a combination (x, y) of random variables be $\phi_f(x, y)$ and a vector including this in its elements be $\Phi(x, y)$, where x is an input symbol for the conditions of a model and y is a label that the model outputs. Let the significance of feature f be represented by θ_f and a vector including θ_f in its

elements be Θ. Then the degree of confidence of giving a label can be expressed by equation 3.4.

$$\langle \Theta, \Phi(x, y) \rangle = \sum_{f \in F} \theta_f \phi_f(x, y) \tag{3.4}$$

Using this, let equation 3.5 defines a conditional probability $P(y|x)$. This is an expression directly to represent the probability model of a CRF.

$$\Pr(y|x) = \frac{\exp\langle \Theta, \Phi(x, y) \rangle}{\sum_{y \in Y} \exp\langle \Theta, \Phi(x, y) \rangle} \tag{3.5}$$

where Y is a set of labels. A label can be forecasted by equation 3.6.

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \log \Pr(y|x) \tag{3.6}$$

In the definition expression of $Pr(y|x)$, the denominator does not depend on y. Thus the result of forecasting based on the equation above is equal to the result of forecast based on equation 3.7.

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \langle \Theta, \Phi(x, y) \rangle \tag{3.7}$$

As seen here, a CRF is in a form in which all of the input symbol x of a model can affect the estimation of an output label. In an estimation with a model, their parameters $\hat{\Theta}$ are computed using equation 3.8 such that maximize the likelihood of the model for given learning data.

$$L(\Theta) = \prod_{i=1}^{N} \Pr(y^{(i)}|x^{(i)}; \Theta) \tag{3.8}$$

With a log likelihood, equation 3.8 can be transformed to the equations 3.9 and 3.10.

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \Pr(y^{(i)}|x^{(i)}; \Theta) \tag{3.9}$$

$$= \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \frac{\exp\langle \Theta(x^{(i)}, y^{(i)}), \Phi(x^{(i)}, y^{(i)}) \rangle}{\sum_{y \in Y} \exp\langle \Theta, \Phi(x^{(i)}, y) \rangle} \tag{3.10}$$

The equations 3.11 and 3.12 give partial differentials with Θ of the log likelihood of model.

$$\frac{\partial \sum_{i=1}^{N} \log \Pr(y^{(i)}|x^{(i)};\Theta)}{\partial \Theta} = \sum_{i=1}^{N} \left(\Phi(x^{(i)}, y^{(i)}) - \frac{\sum_{y \in Y} \Phi(x^{(i)}, y) \exp\langle \Theta, \Phi(x^{(i)}, y) \rangle}{\sum_{y \in Y} \exp\langle \Theta, \Phi(x^{(i)}, y) \rangle} \right) \tag{3.11}$$

$$= \sum_{i=1}^{N} \left(\Phi(x^{(i)}, y^{(i)}) - \sum_{y \in Y} \Phi(x^{(i)}, y) \Pr(y|x^{(i)};\Theta) \right) \tag{3.12}$$

By applying various numerical methods to the equations 3.11 and 3.12, the parameters of model can be obtained.

3.4.4 Experimental settings

To evaluate the validity of the proposed technique, we conducted an experiment to extract question segments and identify question types, using actual question articles. For experimental data we chose 954 queries from 2234 queries in corpus given tags for question types obtained as a result of question type annotation in Section 3.3, excluding the articles to which condition a), b) and c) below applies.

a) The queries include the Yes–No type or other types.

b) The queries include sentences that have different question types in one sentence.

c) The queries do not include a question describing in multiple unadjacent sentences.

As the Yes–No type can turn into whatever type as an answer, it need to handle in a different way from that when handling other question types.

Hence we decided not to include the Yes–No type in our present study. Since the questions that include more than one question in a sentence require preprocessing not directly involved in sentence chunking, those are not covered in the present study, either. The definition of the question segment introduced

Table 3.5. Transition of Question Type in Adjacent Sentences in Question Segments.

Portion	Transition	Frequency
Border	Yes	94 / .53
	No	58 / .32
Inside	No	27 / .15
Total		179

at a question type annotation experiment in Section 3.3.3 indicates that there is no guarantee that a question segment can comprise adjacent sentences only. Actually, in the results of question type annotation we obtained, the unadjacent sentences annotated for a same queries are more than one case. Since such cases were few, accounting for only 2 percents of the whole and our corpus does not contain many learning samples, it is preferable to handle unadjacent sentences as a different chunk than handling them as a chunk. Hence, the experiments in this thesis eliminate queries comprising questions describing in unadjacent sentences. Sentences were segmented with periods only. No separate processing was given to parts in parenthesis. One single sentence was given one question type. As in the question type annotation experiment in Section 3.3.3 a question type of a sentence was defined to be the question type of passages in the sentence. For the question types in this experiment we used those proposed at the past QA Workshop [117] and ones with unique tags defined based on the results of the previous studies by Tamura et.al.[141].

Table 3.5 represents combinations of question types in adjacent sentences annotated with tags other than O tag in test dataset. In 179 total pairs of adjacent sentences, the 85 percents of the pairs are located at borders of question segments. The about 30 percents of adjacent question segments are annotated with same question type.

The features for chunking are composed of uni-gram and bi-gram of part of speech. When using the bi-gram of part of speech, it is used along with the uni-gram as features for chunking. After feature selection using the frequency of

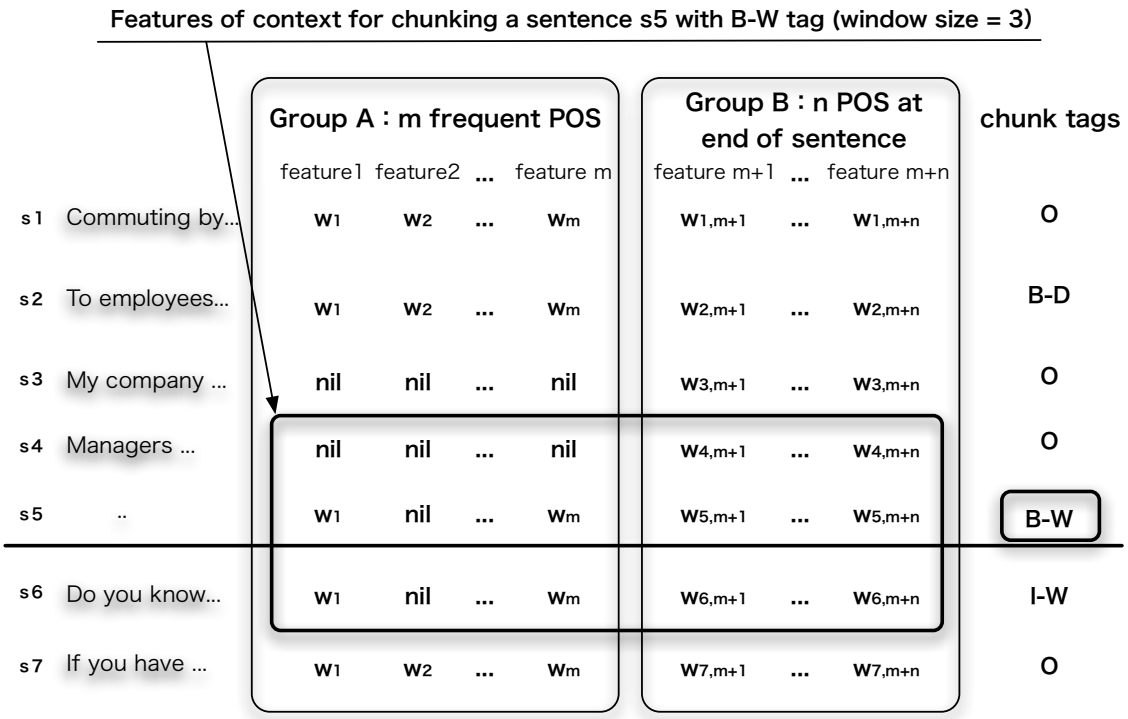

Figure 3.6. Example of data Format in Learning and Testing of Chunking.

features in learning corpus, a thousand of frequent part of speeches are stored. Besides this experiment, the experiment exploiting only several words at beginning and end of sentence are performed. It is the reason why symbols and function words such as question mark and auxiliaries at the end of sentence are expected to be effective for extraction of question segment, and interrogatives at beginning of sentence to work well for question type identification. The number of exploited part of speeches at the beginning and end of sentence varied one to five.

The chunk tag sets comprising four types mentioned in Section 3.4.1 and IO tag set that does not distinct two adjacent question segments, are used for chunking. As the chunker implementing CRF, I used CRF++ supported by Kudo. The learning parameters were set in default values.

Features using this experiment only were combinations of part–of–speech(POS). Uni-gram and bi-gram of POS, and n words from beginning or

[1]http://chasen.org/~taku/software/CRF++/

Table 3.6. Summary of Experimental Settings.

Features	*Set1* : uni-gram of POS	all/content words
	Set2 : uni-gram + bi-gram of POS	all
	Set3 : n POS at the end of santence	n=1-5
	Set4 : n POS at sentence head and end	n=1-5
Chunk tags	IO/IOB1/IOB2/IOE1/IOE2/IOBES	
Window size	one, three and five sentences	

end of sentence were exploited. The number n was varied from 1 to 5. In the case that only uni-gram was exploited, both a feature set only including content words and another set including all words were tested. Figure 3.6 represents the format of feature set of learning and test data for CRF++. It is a matrix of features of sentences. Each column is assigned to one feature and each cell in this matrix indicates a feature value corresponding to the sentence. In this experiment, the values of feature is binary such that are specified by the symbol representing the feature and a symbol indicating absence of the feature.

In Figure 3.6 w_1, w_2, ..., and w_m indicate the top m words in frequent words ranking in the dataset, and $w_{1,m+1}$, $w_{2,m+2}\cdots w_{7,m+n}$ the n words at the end of each sentence. The 'nil' indicates that those features are not included in the sentence.

As the diagram indicates, the feature columns can be divided into several groups of columns and some of groups were exploited in combination. Correspondingly, sentences used as contexts of a targeted sentence for chunking can be selected as same manner. The contexts using in this experiment are only considered in units of sentence, thus we use the idea of "window" of a sequence of sentences as exploited contexts for chunking. The window size varied from only target sentence for chunking through three sentences including one forward and one backward sentence, and to another five sentences including two forward and two backward sentences of the target. Table 3.6 summarizes these experimental conditions.

Table 3.7. Accuracy of Chunking.

	Uni : All	Uni : Content	Uni + Bi : All	#Segments
Accuracy	.29	.18	.29	–
Segmentation	.56	.32	.57	1088
Consultation	.12	.07	.15	66
Description	.3	.11	.34	246
Evaluation	.27	.13	.27	80
Location	.34	.15	.33	108
Name	.34	.20	.30	258
Reason	.33	.06	.35	146
Time	N/A	N/A	N/A	13
How-to	.5	.26	.47	171

The experimental results were evaluated for F-measure on various question types as viewed by question segment. The correct answer rate of chunk identification by a query is computed such that answers being correct in both segment and type are regarded as correct ones. All evaluations were computed in 2-fold cross-validation.

3.4.5 Experimental results

Table 3.7 indicates evaluations of chunking when varying the features for chunking. This is resulted in condition that a thousand of the most frequent words of morpheme in the experimental corpus were used. The value in this table represents F-measure for each question types, and Accuracy presents that regarding a correct case as one correctly assigned the segments and the question types for all questions in a query. These F-measures are independently computed in segment extraction and question type identification. In computation of F-measure of segmentation, the chunking is regarded as correct when it is matched to a correct segmentation.

Accuracies entirely indicate low performance values that mean this task cannot be performed accurately with simple features of words. The accuracies of

Table 3.8. Results of Chunking Varying Window Size.

	window size		
	1	3	5
Accuracy	.29	.28	.28
Segmentation	.57	.57	.60
Consultation	.15	N/A	.03
Description	.34	.33	.32
Evaluation	.27	.17	.20
Location	.33	.22	.19
Name	.3	.28	.28
Reason	.35	.3	.28
Time	N/A	N/A	N/A
How-to	.47	.41	.41

chunking were performed in case using all kind of part of speeches rather than using only content words.

No question segments shows high accuracy regardless of feature selection, but the best performance appears using all part of speeches. Comparing with the results only using uni-gram and additionally using bi-gram, the chunking with bi-grams was performed slightly well than with only uni-grams.

Table 3.8 shows results of question extraction and type identification when varying window size for chunking. The values in the cells of this table were computed as same manner in Table 3.7. As this table indicates, when varying window size of context, no salient difference in accuracy of chunking. However some different changing along with window size in some question types appears in these results.

Table 3.9 presents performances of question extraction using different chunk tag sets. The values in this table indicate F-measures of I/O/B/E/S tags in each chunk tag sets. IO tag set, which cannot recognize adjacent question segments, achieves high F-measure value in type identification of I tag. In IOB1 tag sets, B tag which indicates a boundary of adjacent question segments shows lower performance. In the case of IOB2 tag set, I tag which indicates inside or end

Table 3.9. Question Segmentation with Different Chunk Tag Sets.

	IO	IOB1	IOB2	IOE1	IOE2	IOBES
I	.76	.74	.14	.73	.11	N/A
O	.94	.94	.94	.94	.94	.94
B	-	.16	.74	-	-	.11
E	-	-	-	.13	.73	.15
S	-	-	-	-	-	.72

of a question segment also appears lower performance. This kind of tendency is presented in experimental results of E tag in IOE1 and IOE2. When using IOBES tag sets, S tag of question segment with no adjacent question segment shows high F-measure but the performance of I/B/E tags remains lower.

3.5 Discussion

The results of this experiment did not satisfy our expectation. Especially, the performance of type identification does not far achieve the results in previous studies regarding to single sentence question. In question extraction, the F-measure indicated about 0.6 at most. But this result does not necessarily lead a pessimistic conclusion. For instance, in text summarization many methodologies of text segmentation based-on topic have been proposed. They comprise the studies related to documents with certain document styles, such as news paper, minutes of meeting, papers and patents, in which the accuracies of segmentation shows about 0.7-0.8 in most of the cases. In studies aiming at Web pages and spoken language, accuracy of topic segmentation is even lower. The segmentation in this thesis has to perform question type identification in addition to segmentation of question article from the Web. Nevertheless we used only n-gram of part of speeches in this experiment.

When failing in question segment extraction, the errors often appear in boundaries of adjacent question segments and in the inside of segments comprising two and more sentences. At the boundaries of adjacent segments, by using IOB2/IOE2/IOBES tag sets, the enhancement of performance was recognized.

However when using IOB2/IOE2/IOBES, the performance of labeling the sentence in the inside of a chunk contrarily was declined. The number of this kind of chunks is few in our corpus, the positive examples of this case for machine learning are considered to be insufficient.

There are seventeen question segments comprising multiple sentences in test dataset. The sentence representing question or request appears at the head of segment in one case, at the tail of segment in nine cases and at both the head and tail of segment in six cases. One case has no sentence representing question or request. Those question segments failed to identify the question types. In evaluation of labeling to sentence, the best result was obtained in IOE1 labeling such that four sentences were correctly labeled at 34 heads and tails of sentences of 17 question segments.

This thesis proposed the chunking-based question analysis that performed concurrently both question segmentation and question type identification, which aimed at concurrently solving two problems in question analysis. The first problem was a methodology that can handle more complex queries that comprise multiple questions or question described by multiple sentences, and the second problem is to reduce the computational cost of previous techniques. Proposed methods can solve these problems in theory, however the accuracies in experimental results have not achieved to the practical level yet.

The experimental results show the opposite natures to same features in question segmentation and question type identification. In general, it should be difficult to reveal such two alien problems in a same computational model. Proposed method has not been considered in this aspect of problem. Concurrent processing of question segmentation and question type identification is effective in reduction of computational cost, that however was clarified that does not fit the condition involved different properties of question segmentation and the type identification. Therefore, I am going to change the strategy to that exploiting different models for question segmentation and question type identification in next step, and attempt to reduce the computational cost in such frame work.

Another important observation in experimental result is that many errors of question segmentation and type identification occurred in sentences comprising many ellipses. That process that identify ellipsis and complete it by any relevant

element of sentence, called anaphora resolution [52, 53, 64, 156], is generally difficult, then have not been achieved enough high accuracy to be able to used in practical tasks. As an alternative to avoid anaphora resolution, it is considerable to chunk additional sentences possibly including elided elements. In this point of view, I will enhance question segmentation and question type identification as in following paragraphs.

In question segment extraction, the portion and structure of question segment in a query have not been identified before the processing, thus bag-of-words approach only using words in the query and hypothesizing no question type is plausible. However if question segment comprises many ellipses, the approach only using bag-of-words is not enough to extract features of question segments. As a enhancement to solve this problem, it is considerable to perform only accurate ellipsis analysis over the entire query as preprocessing of chunking.

In experimental results of question type identification, the performances in condition using only features of a chunked segment present better evaluation values than using features of contextual sentences before and after the chunked sentence together. Thus it is considered that it is difficult to improve the accuracy of question type identification by simply adding contexts of chunked sentence. On the other hand, because existence of ellipsis in chunked sentences is problem in question type identification as well as in question segment extraction, any solution of this problem is required. As already shown in previous paragraphs, anaphora resolution conducts not enough accurately in the current technology. In this kind of condition, the solution has to select approaches that acquire any information about elided elements even if anaphora resolution fails to identify those elements. As an expectable way, instead of completely identifying each ellipsis in question segment, selecting chunks involved elided elements and merging features in the chunks to that of target sentence can be considered. Moreover, by using chunking result, it can be possible to remove redundant sentences in a query from search space to identify elided elements.

3.6 Related work

Identification of the question types of question sentences has often been made by pattern matching using lexico-semantic patterns that consider grammar and word meaning classes. A similar strategy has been applied to many other question answering systems since the success of this method in question analysis in early studies of open-domain question answering [24, 68, 117, 154].

For studies using machine learning, techniques based on learning algorithms such as a decision tree [168], a maximum entropy model [59], SNoW [80], and Support Vector Machines [128, 166] have been proposed. In Support Vector Machines (SVMs) [153], Suzuki proposed a question type identification technique using the N-gram of words and their meaning classes as features. The reports of Suzuki indicate that SVMs can bring about the best result of question type identification of a number of conventional learning algorithms such as the decision tree and maximum entropy model.

The previous studies on question-answering in which multi-sentence queries are the input include a study on the classification of sentences included in question answering logs accumulated at the call center of a business [60, 178], a study on automatic answering at the help desk of an academic organization [70, 77], and a study on QA articles at question answering sites on the Internet [141, 185].

Tamura et.al. extracted questions from multi-sentence queries in articles at question answering sites on the Internet and tried to identify question types of these questions [141]. Tamura et.al., expanding their initial method, proposed a technique applicable to cases including more than one sentence in a single article [185]. Their technique, however, depends on manual work for type identification, though question sentences (core sentences) are automatically extracted, and thus it is unclear how accurately it can identify question types in a question article including more than one question.

Tamura et.al technique and the technique we propose here differ in the following points as well: whereas Tamura et.al technique targets questions consisting of a single sentence when extracting question segments, ours can extract questions from a multi-sentence query; in our data of question type annotation is performed with any strings whereas their technique tags only sentences. Since our technique is designed to permit question type annotation of more than one passage for the

same question, it provides tags to be used to associate such passages.

3.7 Conclusion

Through this chapter, we dealt with question segmentation and type identification for multi-sentence queries comprising multiple questions. To sum up, the main contributions are:

- Proposition of new question segmentation and question type identification technique that is advantageous in cost of computation and annotation of corpus, compared with the preceding studies as question segment extraction and question type identification. Our methodology can carry out segmentation and identification at the same time using only one chunker.
- Proposed techniques can handle questions where more than one sentence is required to identify a question type.
- Can identify question types even if more than one question is included in a single article.

Chapter 4

Categorization of Descriptive Answers

4.1 Introduction

In research on question-answering, the question types are often defined as the question focus content type. Therefore, many studies discuss the answering types in the framework for treating question focuses. However, it is also possible to perform different analyses by isolating the answer as a single text from the question.

For factoid type questions, the answers are generally classified by the surface features of the words and phrases and by the semantic classes. On the other hand, for the questions that require answers *descriptive answers* that are described in sentences and texts, it is possible to consider the classification by types of sentences and discourses. Contrary to the factoid type question-answering in which the answer is mainly indicated with words and phrases such as names and quantity, non-factoid question-answering that expects a descriptive answer such as definitions, reasons, reputations, and methods has various forms of description including sentences and texts according to the contents of the answer. Let us suppose that there is a question asking the reputation of dish Y at restaurant X, and the answer is "Dish Y of restaurant X are delicious." This is indeed an evaluation but it is impossible to judge the objectivity of the answer from the answer alone. It is under-specified for users who want to evaluate dish Y of restaurant X. In this case, it is necessary to clarify which descriptive characteristics the question

expects to be evaluated in the answer.

This chapter introduces a leading study on question-answering that expects descriptive answers and another leading study on the classification based on the discursive features of the descriptive answer. Then there follows a report on an experiment automatically categorizing descriptive answers from actual Q&A articles in a Web service based on an analysis of discursive features.

In the following section, I introduce related work of descriptive answers. Section 4.3 presents the result of description type annotation to answers in actual web question-answering service. Subsequently, Section 4.4 describes the result of categorization based on description type of answer. Using machine learning, I explore feasibility of automatic categorization and effective features specifying description types. Finally, Section 4.5 discuss limitation of my approach and the next steps and summarize contributions in this chapter.

4.2 Related work

4.2.1 Question requiring descriptive answers

I have learned from experience that there are more questions that lead to answers described with sentences and texts than those to answers with a few words. The survey of Q&A articles conducted in this study also indicated a high frequency of descriptive answers (cf. Section 4.3). There are some leading studies that call a descriptive answer a "long-answer" because it is composed of long texts rather than words and phrases, and an answer of words and phrases a "short-answer." [13] They also focus on the descriptive features of the answers.

It is not easy to precisely define a descriptive answer and make an exhaustive list of all description types that belong to the class of such answers. Some question types that require a descriptive answer have been proposed, such as the Definition, Reason, Reputation, Opinion, Method, and so forth. When describing answers to these questions, many facts are listed to give definitions and reasons, and the procedure is itemized, which results in a description that tends to be composed of several sentences and longer than the answers to other types of question.

In recent years, I have seen many papers on questions asking definitions [13,

30, 77, 154] and those asking reputations and opinions [58, 71]. The number of papers on reasons [55, 57] and methods [6, 40, 79, 135–137, 184] is increasing, but there are still not many.

Since descriptive answers often consist of several sentences, it is possible to classify the answers by their discursive features and explanatory strategy, to which the conventional general discourse analysis method can be applied.

4.2.2 Discourse analysis

Discourse analysis has a long history, is very extensive, and encompasses many study cases. The scope extends from the analysis of natural interaction [49, 180, 181] to that with literal "reading" [189]. Here, I introduce cases that deal with explanatory written texts. Textual discourse analysis identifies text segment types such as clauses, sentences, and paragraphs, and the logical and rhetorical relations among them [16, 45, 46, 55, 57, 65, 177, 183, 190]. The Rhetorical Structure Theory(RST) [16] is one of the most often used discourse analysis methods in natural language processing. Mann et al. built a bottom-up dependency tree called a rhetorical structure by defining logical and rhetorical relations between clauses and fixing the dependency among the clauses. Based on their idea, Marcu et al. proposed a method for automatically generating a rhetorical structure tree from the corpus [85]. Rhetorical structure tags based on RST have been appended to some large corpora [110].

Some previous work studied Japanese corpus annotation based on description type. For instance, in annotation by human, there are categorization of definition of word in dictionary [150], annotation of causal relation between sentences [57], and in annotation by computer, automatic tagging to definition statements of words in web pages [31]. Those work mentioned problems of this kind of annotation as follows;

- Huge amount of corpus are required to prove statistically any hypotheses, because the number of annotated tags for description types per an article are relatively a few comparing other linguistic annotation.
- Low efficiency of annotation due to read the long context of expression when assigning a tag to the expression.

- Varying annotated expressions so that cannot be acquired rules state description types. Therefore it is necessary to gather automatically corpus and to extract features of description types.

Some interesting leading studies have been conducted on discourse analysis on the Japanese language [51, 95, 170, 175]. However, for actual answer corpus of question-answering in Japanese, previous work is merely found. Maynard [88] explored the structures of answers in Q&A of radio programs and tried to typify them.

4.2.3 Answering procedures

In recent open domain question-answering, I have seen many studies that responds with definitions, reasons, and reputations. However, there have been only a few leading researches on question-answering that responds with methods. Studies on method retrieval with limited text styles and domains such as searching for patents [32, 122] and cooking recipes [40, 121, 125] have been conducted for a long time. Questions related to all procedures were addressed by an expert system [9]. However, only a few studies have been conducted on question-answering that responds by searching for methods from an open domain text set such as Web texts [5, 135–137, 163]. Additionally, such kind of question-answering system requires a more flexible and more machine-operable approach because of the diversity and changeable nature of the information resources. 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 sense, the requirements for categorization in such applications are different from those in previous categorizations. In text categorization research, feature selection has been discussed [120, 130, 132, 162]. However, most of the research dealt with categorization into taxonomy related to domain and genre. The features that are used are primarily *content words*, such as nouns, verbs, and adjectives; functional words and frequent formative

elements were usually eliminated. However, some particular areas of text categorization, for example, authorship identification, suggested the feasibility of text categorization with functional expressions on a different axis of document topics [63, 147, 187].

4.3 Annotating description types of answers

As stated at the beginning of this chapter, the classification of answers has often been discussed as it is integrated with the classification of questions. However, there are no established categories of descriptive answers, and the relationships between classification categories and question categories have not been clarified either. Therefore, I conducted an experiment to classify answers using the classification categories based on the discursive features on general texts that were proposed in leading studies. The classification was performed by tagging the answer articles. I tried to clarify necessary conditions for categories of descriptive answers and those tagging methods.

4.3.1 description types

To further explore description types of answers, this thesis considered the framework to solve four problems comprising those described in last section. For the first problem of collection of corpus for annotating description type and the second problem of reduction of annotation cost for tagging, this thesis suppose a network environment for anonymous annotators tagging descriptive types to articles. To realize such kind of annotation framework, at least, I have to know any description types that can be stably assigned by non-professional annotators. This thesis supposed instructions of annotations and definitions of description types in a level of book of technical writing for general readers, and then investigated the feasibility of annotation in such kind of discursive features of text. For the third problems, this thesis stands on machine learning based approaches to automatically acquire rules to specify descriptive type from tagged corpus. Finally, for the forth problem of feature analysis for answers in Japanese question-answering, I conducted annotation of description types to answers in a actual web Q&A service, and examine the features of description types.

Based on definition of types in paragraph writing by Shinoda [175], I defined eleven category types Table 4.1 for classifying answer articles. During the classification, some typical example articles applicable to each category were presented to the subject (see Appendix B). Shinoda suggested "development of explanation within paragraphs in technical documents" This includes logical, rhetorical, and discursive relations among various sentences and clauses.

As discussed in Section4.2, many classification categories for text discourse types have been proposed. However, there is no category that is appropriate as a standard. Numerous studies have been conducted on bottom-up representation of discourse structure [16, 45, 65] but such categorization requires a relatively high level of linguistic training. In fact, it was reported that inconsistencies exist among subjects who were thought to have adequate linguistic training [140].

Recently, mechanisms that assume tagged Web documents such as Semantic Web and social bookmarking are being used on the Internet. These are beginning to form a group of contents called Consumer Generated Media (CGM); such massive tagging mechanisms have not existed before. In such a tagging environment, the tagging schemes that have been used by language processing specialists are difficult to implement due to skills and work time. I therefore used classification categories for text creation that is written for the general public. I surveyed what stability can be expected in classifying discursive types using categories that do not assume sophisticated linguistic training.

4.3.2 Annotation environment

The view window of annotation tool consists of four components as shown in Figure 4.1, which are text pane, check boxes or pull-down menus to select categories to annotate, status field for management information like article ID, and article selector.

The annotator reads answer articles showed in the text pane, and then categorizes them using the check boxes and pull-down menus to select a suitable type for a question. The definitions of categories and procedures of annotation can always be referred to from the view window of the annotation tool.

For most of the answer articles in a dataset, the annotators can see the whole article in one view, but can also use the scroll function to browse a long article.

Table 4.1. The Definitions of Description Types.

	Description type	Definition
1	*Analysis*	Enumeration of the theoretical connections between all the concepts that were first shown in the general discussion. Then these enumerated concepts will be analysed in terms of fundamental elements, levels and ideas. The hierarchical relationship will be explained.
2	*Fact*	To accumulate facts little by little. To use these facts to support, verify and amplify the general content of the text.
3	*Instance*	Showing a fact as under 2 with a concrete example.
4	*Definition*	Show the definition, then use some facts to demonstrate the definition. Only a definition is also ok.
5	*Order of time*	Record the order in which things happened.
6	*Process*	Show the method of some functional process or the movement of some object.
7	*Conclusion-reason*	First say the conclusion, then name the propositions of the conclusion in order of importance.
8	*Phenomenon-problem*	After explaining some phenomena or facts, then explain their problems or reasons.
9	*Cause-result*	Consisting of cause-result.
10	*Problem-solution*	First explain the problem, then show one by one the solutions in the order of importance or in the order of interest for the reader.
11	*Comparison*	Compare two or more phenomena.

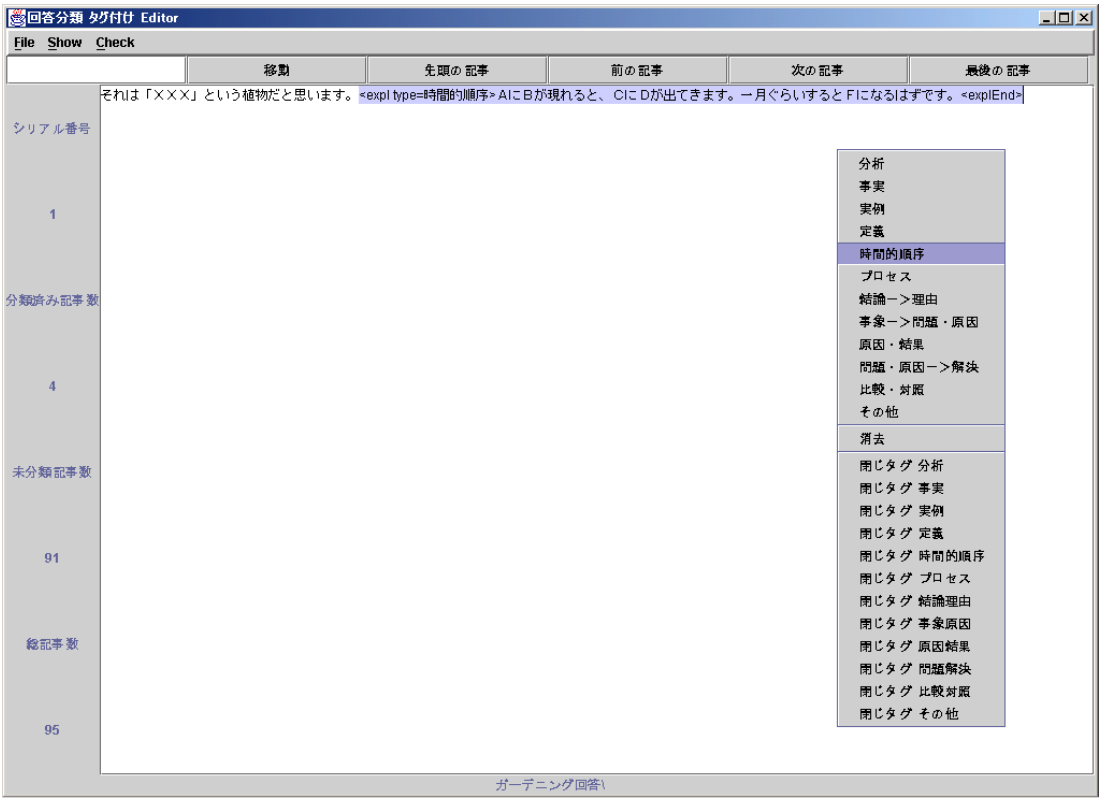

Figure 4.1. Annotation Tool for Answer Articles.

The annotators can start the annotation at any article in the dataset and can change the result of the annotation at any time. The datasets with my experiments are composed of answer articles and corresponding question articles, but the question articles were not presented to the annotators. When annotators put an annotation on an article, the tool logged the time stamp at that point in time.

The annotators could interrupt and resume their annotation in my experiments. They annotated the articles independently in different places and all discussion of judging articles was prohibited. The annotators were also not permitted to annotate answer articles by referring to the question articles.

Before annotating the articles, the rater was given examples of the results of annotating some typical answer articles (see Appendix B).

4.3.3 Overview of datasets

Table 4.2 shows an overview of the article set used in the experiment. Sentence length is presented by the number of bytes.This article set is the same as that of the Q&A that includes the question articles used in Chapter3. Only the answers were extracted from the Q&A set and used in the experiment.

Table 4.2. The Answer Dataset from Six Categories of Oshiete! goo.

	#documents	Ave. #sentences per doc.	sentence length
Gardening	95	7.1/5.7	73.0/59.7
Economics	99	5.6/5.1	85.4/53.4
Healthcare	136	6.1/4.8	82.3/69.4
Politics	168	9.3/9.9	92.5/63.8
Law	132	6.0/5.1	81.4/56.6
Society	150	7.4/5.7	85.8/58.9

4.3.4 Type annotation results

For this article set, each of two language-tagging experts tagged all articles. Although I consider a discursive type of text for non-professional annotators, firstly started this study with professional annotator to test the stability of the set of discursive types. Their tasks were to read the answer articles, select description type used in the articles, and enclose applicable places with a pair of tags. Each expert read the articles one by one and tagged the parts that they thought applied to the description type in the selection. It was also allowed to tag several description types at a single place of the text. If an expert thought that an article did not apply to any description types, he or she could add the Others type.

Table 4.3 shows the tags assigned by the two experienced language-tagging experts, summarizing by types and article domains. The numbers in the Table 4.3 indicate the rate of the frequency of tags in each category to the all tags. All of the six domains show that the Fact type and Instance type occupy more than half of all, suggesting a heavy bias of certain tag type. The number of tags other than these two tags indicate low frequencies in all domains.

Table 4.3. The Result of Description Type Annotation.

	Gardening	Healthcare	Economy	Society	Politics	Law	Total
Analysis	4/.01	8/.02	3/.01	3/.01	5/.01	0/.00	23/.01
Fact	94/.26	108/.25	103/.42	164/.37	199/.36	158/.43	826/.34
Instance	92/.25	102/.24	52/.21	106/.24	87/.16	58/.16	497/.21
Def[1]	4/.01	6/.01	19/.08	8/.02	13/.02	4/.01	54/.02
OoT[2]	0/.00	0/.00	1/0.00	4/.01	21/.04	0/.00	26/.01
Process	34/.09	2/.00	2/.01	10/.02	0/.00	20/.05	68/.03
Co-Rea[3]	15/.04	32/.08	2/.01	14/.03	44/.08	38/.10	145/.06
P-P[4]	3/.01	9/.02	1/.00	4/.01	4/.01	3/.01	24/.01
Ca-Res[5]	15/.04	20/.05	9/.04	25/.06	46/.08	25/.07	140/.06
P-S[6]	51/.14	55/.13	7/.03	41/.09	44/.08	31/.08	229/.10
Comp[7]	24/.07	27/.06	6/.02	11/.02	41/.07	5/.01	114/.05
Others	25/.07	57/.13	43/.17	54/.12	52/.09	27/.07	258/.11
Total	361	426	248	444	556	369	2404

The rate of the assigned description types is similar in each domain. However, there are some description types in which the frequency is distinctively high compared to the other domains such as the definition type in the economic domain, and process types in the gardening domain and legal domain.

I evaluated the agreement between two tagging experts using kappa statistics. As in this experiment, the experts could place several tags, and a kappa value was calculated for each tag type. The kappa statistics was described in next section.

[1]Definition
[2]Order-of-Time
[3]Conclusion-Reason
[4]Phenomenon-Problem
[5]Cause-Result
[6]Problem-Solution
[7]Comparison

Table 4.4. Categorization of n Objects and m Categories.

	Category						
Object	1	2	...	j	...	m	
1	n_{11}	n_{12}	...	n_{1j}	...	n_{1m}	S_1
2	n_{21}						S_2
⋮				⋮			⋮
i	n_{i1}		...	n_{ij}	...	n_{im}	S_i
⋮				⋮			⋮
n	n_{n1}		...	n_{nj}	...	n_{nm}	S_n
	C_1	C_2	...	C_j	...	C_m	

4.3.5 The evaluation of agreement using Kappa statistics

Kappa statistics has been used in previous studies in discourse analysis and text summarization [18, 126, 140]. The kappa value is defined by a formula 4.1 which means subtracting chance agreement from observed agreement.

$$kappa\ value = \frac{P(A) - P(E)}{1 - P(E)} \tag{4.1}$$

Here, $P(A)$ is the proportion of times that the annotators agree and $P(E)$ is the proportion of times that I would expect the annotators to agree by chance. The kappa value is found as determined by the following process below. Consider a dataset of n question articles, each of which is to be assigned to one of m question types. Each of a group of k annotators classifies each article into a question type. The assignments would be represented in Table 4.4 where n_{ij} is the number of annotators assigning the ith article to the jth question type. The total frequency in each row is equal to k.

Let C_j be the number of times that an article is classified into the jth category. This is the column sum of frequencies which can be denoted by $C_j = \sum_{i=1}^{n} n_{ij}$.

To find P(E), note that the proportion of articles assigned to the jth category is $p_j = C_j/n \bullet k$. If the annotators make their assignments at random, the expected proportion of agreement for each category would be p_j^2, and the total expected agreement across all categories can be computed by equation 4.2

Table 4.5. Evaluation of Kappa Value.

Evaluation	Kappa value
Not Good	.00 – .40
Moderate	.41 – .60
Substantial	.61 – .80
Near perfect	.81 – 1.00

$$P(E) = \sum_{j=1}^{m} p_j^2 \tag{4.2}$$

The extent of agreement among the raters regarding the ith article is the proportion of the number of pairs for which there is agreement with the possible pairs of assignments. For the ith articles, this is computed by equation 4.3.

$$S_i = \frac{\sum_{j=1}^{m} \binom{n_{ij}}{2}}{\binom{k}{2}} \tag{4.3}$$

To obtain the total proportion of agreement, I find the average of these proportions across all articles rated using equation 4.4.

$$P(A) = \frac{1}{n} \sum_{i=1}^{n} S_i \tag{4.4}$$

Table 4.4 summarizes the criteria used to evaluate kappa values.

It was impossible to obtain a kappa value for each domain because there were description types with low frequencies as shown in Table 4.3. Instead, I calculated the agreement of tagging by totalizing the six domains based on the same summarization and found moderate levels of agreement for the Definition and the Process types, and substantial level of agreement for the Order of time. There was also certain agreement in the Instance type in the Gardening and the Social domains, and the Comparison type in the Healthcare and the Political domains. For other combinations, no agreement was found or evaluation was impossible because of low frequencies.

4.3.6 Discussion

In the field of natural language processing, studies on tagging have been conducted for a long time. As the corpus-based method became the mainstream, tagging and corpus creation have long been discussed. In general, dictionaries and thesauruses for natural language processing including parsers and taggers, semantic analysis, and discourse analysis are made by specialists. Precise and large dictionaries and thesauruses are indispensable for obtaining high accuracy in various processing. In addition, it must be possible to extend and modify the dictionaries to introduce new words and analysis methods. However, the creation and maintenance of dictionaries and corpus by specialists are costly, and ways of solving this problem are often discussed.

Tagging discourses and tagging for context processing, in particular, often require reading a sizable amount of texts even if only a few tags are placed. In this kind of tagging which cannot obtain much from a single article, the problem of cost of corpus annotation is more serious in the case of machine learning.

In the field of the Internet, there are some researches such as the Semantic Web which pursue more intelligent retrieval and applications that assume annotation by users other than linguists and language processing specialists. In such a tagging paradigm, precise tagging is expected to be very difficult. However, there is a possibility of solving constantly-discussed problems such as high tagging costs and the rapid introduction of new words.

In view of the above, this study tried tagging using discourse tags based on school education and general text creation, rather than discourse tags that require linguistic training used in conventional language processing. I found that there are some discourse types that indicate relatively high levels of agreement for Q&A articles even in the method in which the definitions and examples of the description types were simply taught. Specifically, relatively good agreement was obtained in the Definition, the Order of time, the Process, the Instance, and the Comparison description types. I could not derive a statistical result because the number of articles was limited. However, my results suggest a direction for future research on question-answering that requires descriptive answers.

4.3.7 Problems of answer annotation

Finally I will point out some issues concerning the classification and tagging of descriptive answers.

First, I obtained relatively good agreement in the Definition, the Order of time, the Process, the Instance, and the Comparison description types, but the actual agreement was not high enough. I expect that tagging accuracy will be improved as more detailed studies are conducted on each description type. However, a certain size of fluctuation in tagging is unavoidable as long as I are pursuing tagging by a group of non-professional annotators with various levels.

It is necessary to review what mechanisms of agreement are possible and where the final answer should be sought assuming tagging fluctuations. There have been some studies of this type, albeit few in number [129].

Secondly, there is an issue of data sampling. The data set collected for this study contained only a few discourse types in some domains, and there have been few surveys on such bias. However, a similar tendency can be expected on other Q&A articles of the same kind looking at the research on question types by Tamura et al. [141]. Therefore, for future data sampling, an essential issue is how to prepare a sufficient amount of data and exclude the dependency on specific domains of an experiment.

There is no question about the need for precise language resources. To obtain these, tagging by linguistic and language processing specialists will continue to be required in the future. However, once reliable grammar, rules, and lexical knowledge are described, and they can be used continuously without major change, it will not be necessary to use tags with great fluctuations. Tagging by non-professionals can be applied in cases where dictionary generation is costly relative to performance requirements, the application is personal or in small projects that the cost of creating language resources is not affordable. I think that both professional and non-professional methods will complement each other.

4.4 Description type based answer categorization

In preceding sections, the study to description types of answer in actual question-answering showed that three types of description types, that are the Definition, the Order of time and the Process, can be annotated in high inter-annotators agreement. In the rest of this chapter, this thesis discussed text categorization based on the three description types. The interests mainly fall upon the accuracy and extraction of more features of description types.

To achieve this aims, I exploited a text categorization tool, iSort [152] for experiments. It is a learning-based categorization tool and automatically learns weights of rules for categorization from learning corpus. The rules based on words and their co-occurrences are automatically acquired as word, phrase and sentence patterns with the weights determined by frequency or Kullback-Leibler divergence [25]. Kullback-Leibler(KL) divergence is often used when measuring a distance between two probability distributions. Let $P(x)$ and $Q(x)$ be two probability distributions of a random variable x, KL divergence $D(P||Q)$ is defined by equation 4.5.

$$D(P||Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)} \tag{4.5}$$

Using terms of equation 4.5 and defining a weight of word $w_i (i = 1, 2...N)$ in a category C_j, distance between word frequency distribution $p[w_i|C_j]$ in the category C_j and in all word frequency distribution $p[w_i]$ can be modeled. $KL[w_i, C_j]$ for the weight of w_i in a category C_j is defined by equation 4.6.

$$KL[w_i, C_j] = p[w_i|C_j] log \frac{p[w_i|C_j]}{p[w_i]} \tag{4.6}$$

The distance between word frequency distribution in category C_j and that of whole articles can be presented as equation 4.7.

$$KLD[C_j] = \sum_{i=1}^{N} p[w_i|C_j] log \frac{p[w_i|C_j]}{p[w_i]} \tag{4.7}$$

The word rules, phase rules, and sentence rules are computed based on those word weights $KL[w_i, C_j]$.

The other key features of iSort are that effective rules in categorization are presented in comprehensible forms, and that parameter settings for learning algorithm can be user-friendly handled.

4.4.1 Experimental settings

The experiment of categorization based on three description types of definition, order of time, and process is performed in closed test set. Firstly, I extracted tagged parts of answer text with either of three description types, 112 parts, and then make three categories consisting of articles such that a tagged part corresponds to an article. Only words are exploited as features for categorization and examined ten different feature sets according to combinations of the part-of-speeches, that are noun, collocation, adjective, nominal adjective, verb, auxiliary, adverb, conjunction, adnominals, particle and others [149, 169]. For the rules for categorization, seven different combinations of word rules, phrase rules, and sentence rules, were tested. The weighting methods of rules selected Kullback-Leibler divergence.

4.4.2 Experimental results

To evaluate categorization performance, F-measure [7, 69, 84, 145] is calculated with precision (P) and recall (R) in formula 4.8.

$$F = \frac{2PR}{P + R} \tag{4.8}$$

Here, let $|Ra|$ be the number of relevant documents in categories, $|A|$ be the number of categorized documents, and $|Rc|$ be the number of relevant documents in categorized documents. Precision and recall are defined by the equations 4.9 and 4.10 respectively.

$$P = \frac{|Rc|}{|A|} \tag{4.9}$$

$$R = \frac{|Rc|}{|Ra|} \tag{4.10}$$

Varying combination of part-of-speeches, the feature set mainly consisting of functional words shows good result, 0.83 in F-measure when removing noun, collocation, adjective, nominal adjective, and verb from features. Seeing with respect to each the description types, in same combination of POS above, the Process and the Order of time achieved 0.88 and 0.76 respectively. The Definition type resulted in 0.85 when additionally removing auxiliaries.

For rules for categorization, the highest performance of F-measure is 0.79 resulted in the combination of word rules and phrase rules. For each the Description types, 0.87 for the Process type and 0.71 for the Definition using word and phrase rules, 0.72 for order of time only using phrase rule.

When only using sentence rules, the accuracies of categorizations for the Process type remain in high level, more than 0.6, however that for the Definition and the Order of time were declined drastically.For observations of acquired rules in these experiments, for the Process type, combinations of words such as the particle such as "ので (*node*)" describing reason and the auxiliary such as "てください (*te kudasai*)" descriving requests, expressions at terminals of clauses or sentences often appeared. For the Definition types, a particle of topic marker such as "は (*ha*)," brackets and blank characters were obtained. For the Order-of-time, endings of conjugation marking passed tense such as "た (*ta*)," conjunctive particle such as "という (*toiu*)," and conjunctions such as "しかし (*shikashi*)."

4.5 Discussion and concluding remarks

Proposed question-answering system in this thesis is based on answer extractions using their description types, therefore the scope of applicaation is restricted to answers that are preferentially used certain description types such that identified in surface features. For instance, it is difficult to learn from annotated corpus in description types in that inter-annotators agreements were low, such as Analysis, Fact, Instance and Cause-Result. When answers to a question appear with various description types in source documents, such as free-formatted essay, proposed approach should be not work effectively to such question. On the other hand,

good performance results for description types such as Analysis, Fact, Instance and Cause-Result, were reported in previous studies. By improving the definition of description types and annotation methodology, I consider that the scope of application of proposed methods can be expanded.

For three description types of definition, order of time, and process, this thesis clarified the followings;

- Availability of accurate annotation based on description type.
- Accurate automatic categorization based on description types.
- Effectiveness of sentence patterns for categorization of Process type.

Different approaches for each description type are required, because three description types of definition, order of time, and process shows different natures on same feature sets. Finally, experiments in this thesis are performed in a small dataset. To further explore this topic, larger corpora are demanded.

Chapter 5

Extraction of Procedural Expressions

5.1 Introduction

In Chapter 4, I discussed text categorization based on description types and showed some accurately categorized description types such as definition, order of time, and process. In this chapter, I focus on questions requiring a procedure and intend to study the features necessary for its extraction of the answer. In open-domain question answering, especially in user navigation on the Web, 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 these tasks.

In contrast, it has been reported that the texts related to question-answering in web pages contain many lists in the descriptions. I 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 chapter, my goal is to

Table 5.1. Result from a Search Engine.

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

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 does not lose its effectiveness in my study. In the initial step of my work for this type of QA, I discuss a text categorization task that divides a set of lists into two groups: procedural and non-procedural. First, I 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 I 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.

5.2 Answering procedures with lists

I 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 thesis addresses the issue in the context of the solution being to return to the actual procedure. In the initial step of this study, I focused on the case that the continuous answer candidate passage is in the original text and furthermore

Table 5.2. 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

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 as shown in Figure 5.1[1]. These are:

a) a large number of lists in Q&A articles 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.

5.3 Collection of lists from web pages

To study the features of lists contained in web pages, web pages comprising lists were collected as shownin Figure 5.2. The sets of lists were made according to the following steps (see Table 5.1) :

Step 1 Enter *tejun* (procedure) and *houhou* (method) to Google [14] 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*).

[1]This example excerpts from the readme file of software robots Kairai [124].

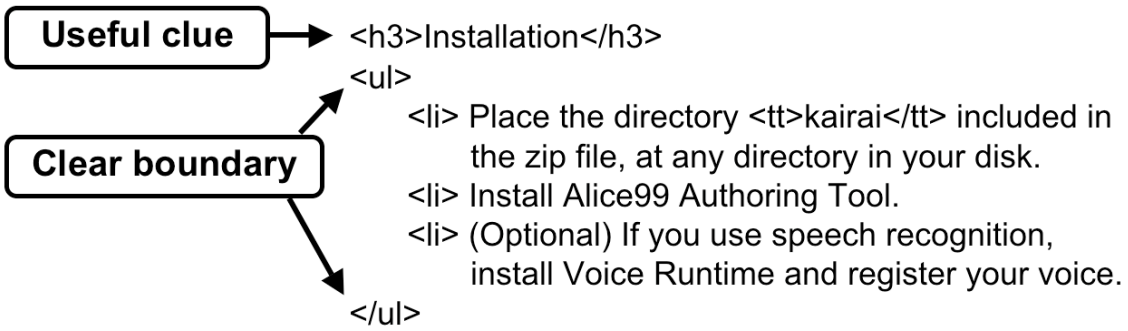

Figure 5.1. Example of Procedural List.

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 [126] is applied to the result. I 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 5.2 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 consisting of only a few documents and were lumped together into a document set named "*Others*." This

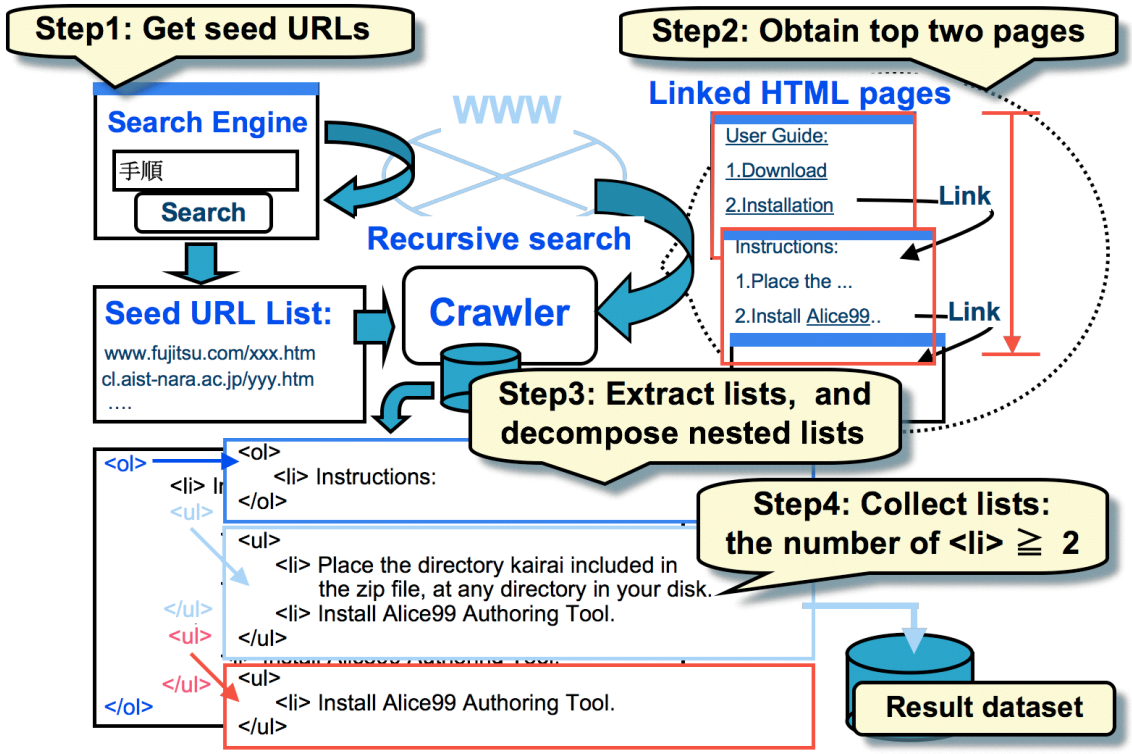

Figure 5.2. Collection of Lists from Web Pages.

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.

5.4 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 description 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 [63, 147, 165]. 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. my approach is based on these features.

5.5 Features : baseline

In addition to the features based on the presence of specific words, I examined sequences of words for my task. Tsuboi et.al.[147] 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 usage of article 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:

[1]Except verbal nouns

[2]Except sentence-final particles

Table 5.3. Types of Tags.

	Tag types	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	adjuctive
	aup	sentece-final-particle
		auxiliary verb
		suffix
	ij	interjection
	seg	others (punctuation, etc.)
	unknown	unknown word

Step 1 By using ChaSen [87], a Japanese POS(Part Of Speech) tagger, I put the document tags and the POS tags into the list. Table 5.3 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, I conducted categorization with SVM. It is one of the large margin classifiers, which shows high generalization performance even in high dimensional spaces [153]. SVM is beneficial for my task, because it is unknown

Table 5.4. 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

which features are effective, and I must use many features in categorization to investigate their effectiveness. The dimension of the feature space is relatively high.

5.6 Features : sequential patterns

Sequential pattern mining consists of finding all frequent subsequences, that are called *sequential patterns*, in the database of sequences of literals.

Besides conventional pattern matching techniques [38], Apriori [2] and PrefixSpan [62] 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 computational cost. The PrefixSpan algorithm succeed in reducing the cost of computation 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 [62].

5.7 Experimental settings

In the first experiment, to determine the categorization capability of a domain, I 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, I 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 5.1 that is same as equation 4.8 in Chapter 4.

$$F = \frac{2PR}{P+R} \tag{5.1}$$

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 5.4 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 5.4 are the mean on the left and the deviation on the right. I employed Tiny-SVM[2] and a implementation of PrefixSpan[3] 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 my experimental conditions.

To investigate the effective features for list categorization, feature sets of the lists were divided into five groups (see Table 5.5) with consideration given to the difference of content word and function words according to my observations (described in Section 5.4). The values in Table 5.5 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 5.3.

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 my observations (described in Section 5.4).

To observe the performances of SVM, I compared the results of categorizations

[2]http://chasen.org/~taku-ku/software/TinySVM/

[3]http://chasen.org/~taku-ku/software/prefixspan/

Table 5.5. 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

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[4], was chosen.

In these experiments, only the first sentence in each list item was used because in my preliminary experiments, I 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, I 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, I selected 100, 300, 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.

5.8 Experimental results

Table 5.6 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, dif-

[4]http://www.cs.waikato.ac.nz/~ml/weka/

Table 5.6. 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 5.7. 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 5.8. 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

ferences 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 did not 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 5.7 and 5.8).

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 5.1, 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 5.9 compares the results of SVM and j48.j48 decision tree. Table 5.10 lists the effectiveness of mutual information filtering.

In both tables, values show the F-measure calculated with formula 5.1. According to Table 5.9, SVM overtook j48.j48 overall. j48.j48 scarcely changes with

Table 5.9. 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 5.10. 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

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.

5.9 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, I can recognize that the simple methods of other pre-cleaning are not notably effective when learning from documents of the same domain. However, the simple methods work well in my 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. I also often found in their expressions that for a particular class of content word, special characters were adjusted (see Figure 5.3).

This type of pattern occasionally contributed the correct classification in my 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.

5.10 Summary

The present work has demonstrated effective features that can be used to categorize lists in web pages by whether they explain a procedure. I show that categorization to extract texts including procedural expressions is different from traditional text categorization tasks with respect to the features and behaviors

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

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

Figure 5.3. Example of Effective Patterns.

related to co-occurrences of words. I 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.

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 my 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.

Chapter 6

Conclusion

I aim to develop fundamental technologies of open domain question-answering that properly process queries comprising multi-sentences and requiring descriptive answers. Although computer systems handle many challenging tasks, in real world, this task is ubiquitously accomplished by manpower. The objective of this study was to establish methodology to realize such kind of question-answering in real world by advanced natural language processing. Towards this end, I explored two different aspects, that are question analysis and answer extraction. In question analysis, I concentrated on extracting question segments and identifying question types. In the answer extraction, I examined the description type of real answer articles in the Q&A site by performing discourse annotation, and then proposed a methodology based on description type to extract descriptive answers in special cases.

In Chapter 3, I discussed question segment extraction and question type identification. I show that this is essential for multi-sentence query processing, and propose a novel efficient method that executes segment extraction and type identification at the same time. My methodology solves these tasks by regarding them as chunking problems. Using a machine learning method of a conditional random field and features of words in a query for my chunker. I obtained the following key findings:

- I propose an new efficient sentence-chunking based technique to concurrently identify the segments and the types in a multi-sentence query.

- It is applicable in case that multiple questions are comprised in a query
- It also accepts a question segment including multi-sentences.

Proposed methods can provide benefits above in theory, however the accuracies in experimental results have not achieved to the practical level yet and further enhancement of methodology is required. The experimental results show the opposite natures to same features in question segmentation and question type identification. So it is necessary to change the strategy to that exploiting different models for question segmentation and question type identification in next step, and attempt to reduce the computational cost in such frame work. Additionally, necessity of anaphora resolution was clarified by error analysis of experiments in Chapter 3. In addition to proposed bag-of-words approach, I consider that have to apply different types of anaphora resolvers into question segment extraction and question type identification respectively that mainly conduct ellipsis analyses. They are performed to the former as pre-processing of chunking and to the latter as post-processing.

In this thesis, I limited the discussion to questions comprising only single questions in each sentence. However, there are many questions including multiple questioners in a sentence, therefore the processing methods require multi-sentence query processing. Although I did not also take up identification of relations among question segments in detail, solving this problem is important for completing the lack of information to answer the question with texts in the query. In this thesis, I address question type identification, which however does not guarantee that I can acquire completed information to answer the questions. I intend to develop methodology to solve these problems in future.

In Chapter 4, I examined question-answering that expects descriptive answers, and classification based on the descriptive and discursive features of the answer are discussed. I tried tagging using discourse tags based on school education and general text creation, rather than discourse tags that require linguistic training used in conventional language processing. I found that there are some discourse types that indicate relatively high levels of agreement for Q&A articles even in the method in which the definitions and examples of the answer types were simply taught. Specifically, relatively good agreement was obtained in the definition, order of time, process, instance, and comparison answer types.

I obtained the following key findings:

- Some expected discription types with high inter-annotator agreement, that are definition, order of time and procedure, were found.
- In these description types, the possibility of accurate annotation by non-professionals was shown.
- Proposition of new techniques of answer categorization based on the description type. The experimental results showed a high accuracy of the proposed method with features of functional words, 0.8 in F-measure, for the three description types: definition, order of time and procedure.

I also pointed out some issues concerning the classification and tagging of descriptive answers as future work. The first task was to develop mechanisms to control agreement and disagreement in discourse tagging. The second task was Q&A corpus balanced in question type. I also needed to consider the mixture of professional and non-professional tagging in discourse annotation.

In Chapter 5, I present effective features that can be used to categorize lists in web pages by whether they explain a procedure. I showed that categorization to extract texts including procedural expressions was different from traditional text categorization tasks with respect to the features and behaviors related to co-occurrences of words. I also showed 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.

I obtained the following key findings:

- When restricting the document structure of answers to a presentation of lists, a moderate accurate extraction of procedural expressions can be performed with sequential pattern mining and support vector machines.
- This method showed a high performance, more than 0.7 in F-measure, when extracting lists of procedural expression.

- For extraction of procedural expressions, functional words and patterns are effective.

In the next step, I need to perform the same task using a bigger data. In this thesis, there are too few lists in each specific domain of the data set within the Others domain to reveal its precise nature. I also have to explore the nested structure of lists and the nominal list type.

In closing, I consider that fully automatic descriptive answer extraction is possible. The next step of this study is making an ensemble of human annotated semantic meta-data and computer extracted features when the computer actually extracts an answer. I aim to explore schemes that more directly exploit human annotation to extract answers that are more relevant for user questions.

References

[1] Takeshi Abekawa and Manabu Okumura. Analysis of Japanese relative clauses. *Journal of Natural Language Processing*, Vol. 12, No. 1, pp. 107–124, January 2005.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rulesr. In *Proceedings of 20th International Conference Very Large Data Bases (VLDB)*, pp. 487–499, 1994.

[3] Hassan Alam, Rachmat Hartono, Aman Kumar, Fuad Rahman, Yuliya Tarnikova, and Che Wilcox. Web page summarization for handheld devices: A natural language approach. In *Proceedings of Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, pp. 1153–1157, 2003.

[4] Hassan Alam, Fuad Rahman, Yuliya Tarnikova, and Aman Kumar. When is a list is a list?: Web page re-authoring for small display devices. In *Proceedings of International World Wide Web Conferences (WWW) 2003*, 2003.

[5] Farida Aouladomar. Towards answering procedural questions. In *KRAQ'05 - IJCAI workshop*, July 2005.

[6] Naoki Asanoma, Osamu Furuse, and Ryoji Kataoka. Feature analysis of explanatory documents for how-to type question answering. In *IPSJ SIG Notes NL-168*, pp. 55–60, 2005. in Japanese.

[7] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[8] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of COLING-ACL'98*, pp. 79–85. Association for Computational Linguistics, 1998.

[9] Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum. *The Handbook of Artificial Intelligence*. Kyoritsu Shuppan, Tokyo, 1989. Japanese Edition Translated by K. Tanaka and K Fuchi.

[10] Regina Barzilay. *Information Fusion for Mutlidocument Summarization: Paraphrasing and Generation*. PhD thesis, Columbia University, 2003.

[11] Adam L. Berger and Vibhu O. Mittal. Query-relevant summarization using FAQs. In *ACL*, 2000.

[12] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific America*, Vol. 284, No. 5, pp. 34–43, 2001.

[13] Sasha Blair-Goldensohn, Kathleen R. McKeown, and Andrew Hazen Schlaikjer. *New Directions In Question Answering*, chapter 4 Answering Definitional Questions: A Hybrid Approach. AAAI Press, 2004.

[14] Sergey Brin and Larry Page. The anatomy of a large-scale hypertexual Web search engine. In *Proceedings of 7th International World Wide Web Conference*, 1998.

[15] Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. Question answering from frequently asked question files: Experiences with the FAQFinder system. *AI Magazine*, Vol. 18, No. 2, pp. 57–66, 1997.

[16] Mann William C. and Sandra A. Thompson. Rhetorical structure theory: description and construction of text structures. Technical Report ISI/RS-86-174, Information Sciences Institute, 1986.

[17] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Block-based web search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 456–463, 2004.

[18] Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistic*, Vol. 22, No. 2, pp. 249–254, 1996.

[19] Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Proceedings of International World Wide Web Conferences (WWW) 2003*, pp. 225–233, 2003.

[20] Nello Christiani and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[21] William W. Cohen, Matthew Hurst, and Lee S. Jensen. A flexible learning system for wrapping tables and lists in HTML documents. In *International World Wide Web Conferences (WWW) 2002: Proc. of the 11th international conference on World Wide Web*, pp. 232–241, New York, NY, USA, 2002. ACM Press.

[22] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Neural Proceedings of Information Processing Systems (NIPS 2001)*, 2001.

[23] Michael Collins and Nigel Duffy. Parsing with a single neuron: Convolution kernels for natural language problems. Technical Report UCSC-CRL-01-01, University of California at Santa Cruz, 2001.

[24] Dan I. Moldovan D., Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, and Vasile Rus. LASSO: A tool for surfing the answer net. In *Proceedings of TREC-8*, pp. 175–184, 1999.

[25] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2000.

[26] Koji Eguchi, Keizo Oyama, Emi Ishida, Noriko Kando, and Kazuko Kuriyama. Overview of the Web retrieval task at the third NTCIR Workshop. Technical Report NII-2003-002E, National Institute of Informatics, 2003.

[27] Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, 2005.

[28] The Japanese Society for Artificial Intelligence. *Encyclopedia of artificial intelligence*. Kyoritsu Shuppan, 2005.

[29] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of The 13th International Conference on Machine Learning*, pp. 148–156, 1996.

[30] Atsushi Fujii and Tetsuya Ishikawa. 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-2001)*, pp. 196–203, July 2001.

[31] Atsushi Fujii and Tetsuya Ishikawa. Extraction and organization of encyclopedic knowledge information using the World Wide Web. *The Transactions of The Institute of Electronics*, Vol. D-II Vol.J85-D-II, No. 2, pp. 300–307, February 2002.

[32] Atsushi Fujii, Makoto Iwayama, and Noriko Kando. Overview of Patent Retrieval Task at NTCIR-5. In *Proceedings of NTCIR-5 Workshop Meeting*, Tokyo, Japan, December 2005.

[33] Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. An overview of Question and Answering Challenge (QAC) of the next NTCIR Workshop. In *Proceedings of the Second NTCIR Workshop*, pp. 144–151. National Institute of Informatics, 2001.

[34] Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question and Answering Challenge (QAC1) question answering evaluation at NTCIR Workshop 3. In *Working notes of the Third NTCIR Workshop Meeting Part IV: Question Answering Challenge*, pp. 1–10. National Institute of Informatics, 2002.

[35] Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. Generation of relative referring expressions based on perceptual grouping. In *Proceedings of the 20th International Conference on Computational Linguistics: COLING 2004*, August 2004.

[36] Takahiro Funasaka, Kazuhide Yamamoto, and Shigeru Masuyama. Relevant newspaper articles summarization by redundancy reduction. In *IPSJ SIG Notes NL-114-7*, 1996. in Japanese.

[37] Suhit Gupta, Gail Kaiser, David Neistadt, and Peter Grimm. DOM-based content extraction of HTML documents. In *Proceedings of the 12th International World Wide Web Conferences (WWW) 2003*, pp. 207–214, 2003.

[38] Dan Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

[39] M.A.K. Halliday and Ruqaiya Hasan. *Cohesion in English.* Hitsuji Shobo, Tokyo, JAPAN, japanese edition, 1997.

[40] Reiko Hamada, Ichiro Ide, Shuichi Sakai, and Hidehiko Tanaka. Structural analysis of cooking preparation steps. *The Transactions of The Institute of Electronics*, Vol. D-II Vol.J85-D-II, No. 1, pp. 79–89, January 2002. in Japanese.

[41] Sanda M. Harabagiu, Marius A.Pasca, and Steven J. Maiorano. Experiments with open-domain textual question answering. In *Proceedings of COLING-2000*, Saarbruken Germany, August 2000.

[42] Koiti Hashida. Global document annotation. In *Proceedings of Natural Language Processings Pacific Rim Symposium '97*, 1997.

[43] Marti A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 9–16, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[44] Tsutomu Hirao. *A Study on Generic and User-Focused Automatic Summarization.* PhD thesis, Nara Institute of Science and Technology, 2002.

[45] Jerry R. Hobbs. Coherence and co-reference. *Cognitive Science*, pp. 67–82, 1979.

[46] Jerry Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul Martion. Interpretation as abduction. *Artificial Intelligence*, pp. 69–142, 1993.

[47] Chiori Hori, Takaaki Hori, Hideki Isozaki, Eisaku Maeda, Shigeru Katagiri, and Sadaoki Furui. Deriving disambiguous queries in a spoken interactive ODQA system. In *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing(ICASSP)*, pp. 624–627, 2003.

[48] Chiori Hori, Takaaki Hori, Hajime Tsukada, Hideki Isozaki, Yutaka Sasaki, and Eisaku Maeda. Spoken interactive ODQA system: SPIQA. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pp. 153–156, 2003.

[49] Akira Ichikawa, Masahiro Araki, Masato Ishizaki, Itabashi Shuichi, Toshihiko Itoh, Hideki Kashioka, Keiji Kato, Hideaki Kikuchi, Tomoko Kumagai, Akira Kurematsu, Hanae Koiso, Masafumi Tamoto, Syun Tutiya, Shu Nakazato, Yasuo Horiuchi, Kikuo Maekawa, Yoichi Yamashita, and Takashi Yoshimura. Standardising annotation schemes for Japanese discourse. In *Proceedings 1st International Conference on Language Resource and Evaluation*, SIG-SLUD-9703, pp. 41–48, February 1998.

[50] Hiroshi Ichikawa, Masaki Noguchi, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. eBonsai: An integrated environment for annotating treebanks. In *Proceedings of Asia Federation of Natural Language Processing*, pp. 110–115, October 2005.

[51] Takashi Ichikawa. *Introduction to style theory for Japanese education*. Education, 1978. in Japanese.

[52] Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Anaphora resolution by antecedent identification followed by anaphoricity determination. *ACM Transactions on Asian Language Information Processing (TALIP)*, Vol. 4, No. 4, pp. 417–434, 2005.

[53] Ryu Iida, Kentaro Inui, Yuji Matsumoto, and Satoshi Sekine. Noun phrase coreference resolution in Japanese based on most likely candidate antecedents. *IPSJ Journal*, Vol. 46, No. 3, pp. 831–844, 2005. in Japanese.

[54] Satoru Ikehara, Masahiro Miyazaki, Akio Yokoo, Satoshi Shirai, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. *Ni-*

hongo Goi Taikei - A Japanese Lexicon, Vol. 5. Iwanami Syoten, 1997. in Japanese.

[55] Kentaro Inui and Atsushi Fujita. A survey on paraphrase generation and recognition. *Journal of Natural Language Processing*, Vol. 11, No. 5, pp. 151–198, October 2004.

[56] Takashi Inui. *Acquiring Causal Knowledge from Text Using Connective Markers,*. PhD thesis, Graduate School of Information Science, Nara Institute of Science and Technology, 2004.

[57] Takashi Inui and Manabu Okumura. Investigating the characteristics of causal relations in Japanese text. In *The 43rd Annual Meeting of the Association for Computational Linguistics, Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*, 2005.

[58] Takashi Inui and Manabu Okumura. A survey of sentiment analysis. *Journal of Natural Language Processing*, 2006. in Japanese.

[59] Abraham Ittycheriah, Martin Franz, Wei-Jing, and Adwait Ratnaparkhi. Question answering using maximum entropy components. In *Proceedings of NAACL-2001*, pp. 33–39, 2001.

[60] Reijirou Iwasaki and Kenji Araki. Important sentence extraction method for automatic generation of business days report for conversation data of call center. In *The 19th Annual Conference of the Japanese Society for Artificial Intelligence*, 2005.

[61] Makoto Iwayama and Takenobu Tokunaga. Probabilistic passage categorization and its application. *Journal of Natural Language Processing*, Vol. 6, No. 3, pp. 181–198, april 1999.

[62] Pei Jian, Han Jiawei, et al. Prefixspan: Mining sequential patterns by prefix-projected growth. In *Proceedings of International Conference of Data Engineering*, pp. 215–224, 2001.

[63] Mingzhe Jin. Authorship attribution based on n-gram models in postpositional particle of Japanese. *Mathematical Linguistic*, Vol. 23, No. 5, pp. 225–240, June 2002.

[64] Megumi Kameyama. *Centering theory in discourse*, chapter Intrasentential Centering: A Case Study, pp. 89–112. Oxford, Clarendon Press, 1998.

[65] Megumi Kameyama. *Discourse and Context*, chapter 3 Discourse analysis: coherence and cohesion, pp. 93–121. Linguistic Sciences 7. Iwanami Publishers., 1999. in Japanese.

[66] Noriko Kando. Overview of the fifth NTCIR Workshop. In *Proceedings of NTCIR-5 Workshop Meeting*, 2005.

[67] Hisashi Kashima and Teruo Koyanagi. SVM kernels for semi-structured data. *Machine Learning*, 2002.

[68] Tsuneaki Kato, Jun'ichi Fukumoto, and Fumito Masui. An overview of NTCIR-5 QAC3. In *Proceedings of NTCIR-5 Workshop Meeting*, Tokyo, Japan, December 2005.

[69] Kenji Kita, Kazuhiko Tsuda, and Masami Shishibori. *Information Retrieval Algorithms*. Kyoritsu Shuppan, 2002.

[70] Yoji Kiyota, Sadao Kurohashi, and Fuyuko Kido. Dialog Navigator : A question answering system based on large text knowledge base. *Journal of Natural Language Processing*, Vol. 10, No. 4, pp. 145–175, July 2003. in Japanese.

[71] Nozomi Kobayashi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Opinion extraction using a learning-based anaphora resolution technique. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05), poster*, pp. 175–180, 2005.

[72] Janet Kolodner. *Case-based Reasoning*. Morgan Kaufmann, 1993.

[73] Taku Kudo and Yuji Matsumot. Chunking with support vector machines. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics*, pp. 192–199, 2001.

[74] Taku Kudo and Yuji Matsumoto. Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000*, 2000.

[75] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of EMNLP 2004*, 2004.

[76] Taku Kudoh and Yuji Matsumoto. Japanese dependency analysis based on Support Vector Machines. In *Proceedings of EMNLP/VLC 2000*, 2000.

[77] Sadao Kurohashi and Wataru Higasa. 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*, pp. 141–149, 2000.

[78] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[79] Yu-Sheng Lai, Kuao-Ann Fung, and Chung-Hsien Wu. FAQ mining via list detection. In *Proceedings of Workshop on Multilingual Summarization and Question Answering (COLING)*, 2002.

[80] Xin Li and Dan Roth. Learning question classifiers. In *COLING2002*, pp. 556–562, August 2002.

[81] Elizabeth D. Liddy. The discourse-level structure of empirical abstracts: An exploratory study. *Information Processing and Management*, Vol. 27, No. 1, pp. 55–81, 1991.

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

[83] Inderjeet Mani and Eric Bloedorn. Summarizing similarities and. differences among related documents. *Information Retrieval*, Vol. 1, pp. 1–23, 1999.

[84] Christopher D. Manning and Hinrich Schutze. *Foundation of Statistical Natural Language Processing*. The MIT Press, 1999.

[85] Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, November 2000.

[86] Kunio Matsui, Hiroshi Tsuda, Kenji Ueda, Yusuke Koizumi, Junichi Toyouchi, and Kosei Fume. Semantic Web:meta-data on the Semantic Web and its usage. *IPSJ Magazine*, Vol. 43, No. 7, pp. 709–750, 2002. in Japanese.

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

[88] Senko K. Maynard. *Discourse Linguistics*, chapter 5, pp. 66–95. Kurosio Pub., 2004. in Japanese.

[89] Kathleen McKeown and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 550–557, 1999.

[90] Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards multidocument summarization by reformulation progress and prospects. In *Proceedings of AAAI'99*, pp. 453–460, 1999.

[91] Kathleen McKeown and Dragomir R. Radev. Generating summaries of multiple news articles. In *Proceedings of SIGIR'95*, pp. 74–81. the Association for Computing Machinery, 1995.

[92] Kathleen McKeown and Dragomir R. Radev. *Advances in Automatic Text Summarization*, chapter Generating Summaries of Multiple News Articles, pp. 381–389. The MIT Press, 1999.

[93] Taniya Mishra, Esther Klabbers, and Jan P. H. van Santen. Detection of list-type sentences. In *Proceedings EUROSPEECH'03*, pp. 2477–2480, 2003.

[94] Dan Moldovan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 33–40, 2002.

[95] Masaru Nagano. *A reveiw of style theory.* Asakura, 1986. in Japanese.

[96] Katashi Nagao and Koiti Hasida. Automatic text summarization based on the global document annotation. In *Proceedings of COLING-ACL'98*, pp. 917–921. Association for Computational Linguistics, 1998.

[97] Katashi Nagao, Yoshinari Shirai, and Kevin Squire. Semantic annotation and transcoding: Making Web content more accessible. *IEEE MultiMedia*, Vol. 8, No. 2, pp. 69–81, 2001.

[98] Makoto Nagao, Satoshi Sato, Sadao Kurohashi, and Tatsuhiko Sumida. *Natural Language Processing.* Iwanami Koza Software Science 15. Iwanami Publishers, 1996.

[99] Hiroshi Nakagawa and Toshihiko Watanabe. Automatic text summarization -as an indispenssable element technology for intellectual activity sunnort-:natural language processing and contents transformation for mobile terminals. *IPSJ Magazine*, Vol. 43, No. 12, pp. 16–20, 2002. in Japanese.

[100] Yoshio Nakao. Thematic hierarchy detection of a text using lexical cohesion. *Journal of Natural Language Processing*, Vol. 6, No. 6, pp. 83–112, 1999.

[101] Hidetsugu Nanba and Manabu Okumura. Towards multi-paper summarization using referencee information. *Journal of Natural Language Processing*, Vol. 6, No. 5, pp. 43–62, 1999.

[102] Tomoyuki Nanno, Suguru Saito, and Manabu Okumura. Structuring web pages based on repetition of elements. In *Second International Workshop on Web Document Analysis (WDA2003)*, 2003.

[103] Tomoyuki Nanno, Yasuhiro Suzuki, Toshiaki Fujiki, and Manabu Okumura. Automatic collection and monitoring of Japanese weblogs. In *Proceedings*

of International World Wide Web Conferences (WWW) 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2004.

[104] Fumihito Nishino, Ryo Ochitani, Atsuko Kida, Hiroko Inui, Wakako Kuwahata, and Minako Hashimoto. Information extraction using top - down pattern analysis. In *IPSJ SIG Notes NL-124-13*, 1998. in Japanese.

[105] Tadashi Nomoto and Yuji Matsumoto. Exploiting human judgments for automatic text summarization: An empirical comparison. *IPSJ Journal*, Vol. 45, No. 3, pp. 794–808, March 2004. in Japanese.

[106] Yoshihisa Ohtake, Katsumi Nitta, Shigeru Maeda, Masayuki Ono, Hiroshi Osaki, and Kiyokazu Sakane. Legal reasoning system HELIC - II. *IPSJ Journal*, Vol. 35, No. 6, pp. 986–996, 1994. in Japanese.

[107] Manabu Okumura and Hidetsugu Nanba. Automated text summarization: A survey. *Journal of Natural Language Processing*, Vol. 6, No. 6, pp. 1–26, 1999.

[108] Nobuyuki Omori, Jun Okamura, Tatsunori Mori, and Hiroshi Nakagawa. Hypertextualization for related instruction manuals using the techniques of information retrieval. *Journal of Information Processing Socity of Japan*, Vol. 40, No. 6, pp. 2776–2784, 1999.

[109] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translation:extracting paraphrases and generating new sentences. In *Proceedings of NAACL-HLT2003*, 2003.

[110] Rashmi Prasad, Eleni Miltsakaki, Aravind Joshi, and Bonnie Webber. Annotation and data mining of the Penn discourse treebank. Technical report, Proceedings of the ACL Workshop on Discourse Annotation, 2004.

[111] Fuad Rahman and Hassan Alam. A commercial Web based digital library for sharing and distributing documents. In *Proceedings of First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, p. 93. IEEE Computer Society, 2004.

[112] I. V. Ramakrishnan, Amanda Stent, and Guizhen Yang. Hearsay: enabling audio browsing on hypertext content. In *Proceedings of the 13th international conference on World Wide Web*, pp. 80–89, 2004.

[113] Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 88–94, 1995.

[114] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, 2000.

[115] Takeshi Sagara, Makoto Iguchi, and Kazunori Fujimoto. Research trend of Web trust. *Journal of the Japanese Society for Artificial Intelligence*, Vol. 21, No. 4, p. 430, 2006. in Japanese.

[116] Erik F. Tjong Kim Sang and Jorn. Veenstra. Representing text chunks. In *Proceedings of EACL 1999*, 1999.

[117] Yutaka Sasaki, Hideki Isozaski, Hirotoshi Taira, Tsutomu Hirao, Hideto Kazawa, Jun Suzuki, and Eisaku Maeda. SAIQA : A Japanese QA system based on a large - scale corpus. In *IPSJ SIG Notes FI-64*, pp. 77–82, 2001. in Japanese.

[118] Manabu Sassano. Virtual examples for text classification with Support Vector Machines. In *Proceedings of EMNLP 2003*, 2003.

[119] Madoka Sato and Satoshi Sato. Automated editing for packaging netnews articles. *IPSJ Journal*, Vol. 38, No. 6, pp. 1225–1234, 1997. in Japanese.

[120] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–47, 2002.

[121] Tomohide Shibata and Sadao Kurohashi. Unsupervised topic identification by integrating linguistic and visual information based on hidden markov models. In *Proceedings of The Joint 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL2006, poster)*, pp. 755–762, 2006.

[122] Akihiro Shinmori, Manabu Okumura, Yuzo Marukawa, and Makoto Iwayama. Structure analysis of Japanese patent claims using cue phrases. *IPSJ Journal*, Vol. 45, No. 3, pp. 891–905, 2004. in Japanese.

[123] Yusuke Shinyama and Satoshi Sekine. Paraphrase acquisition for information extraction. In *Proceedings of Second International Workshop on Paraphrasing (IWP2003)*, 2003.

[124] Yusuke Shinyama, Takenobu Tokunaga, and Hozumi Tanaka. Kairai-software robots understanding natural language. *IPSJ Journal*, Vol. 42, No. 6, pp. 1358–1367, June 2001. in Japanese.

[125] Kiyoaki Shirai and Hiroshi Ookawa. Constructing a lexicon of actions for the cooking domain toward animation generation. In *IPSJ SIG Notes NL-164-21*, pp. 123–128, 2004. in Japanese.

[126] Sidney Siegel and Jr. N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences 2nd Edition*. McGraw-Hill, New York, 1988.

[127] Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of he 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[128] Jun Suzuki. *Kernels for Structured Data in Natural Language Processing*. PhD thesis, Nara Institute of Science and Technology, 2005.

[129] Nomoto Tadashi. Bayesian learning in text summarization. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 249–256, October 2005.

[130] Hirotoshi Taira and Masahiko Haruno. Feature selection in SVM text categorization. *IPSJ Journal*, Vol. 41, No. 4, pp. 1113–1123, April 2000. in Japanese.

[131] Tetsuro Takahashi, Kentaro Inui, and Yuji Matsumoto. Methods for estimating syntactic similarity. In *IPSJ SIG Notes NL-150-24*, pp. 163–170, July 2002. in Japanese.

[132] Hiroya Takamura. *Clustering approaches to text categorization*. PhD thesis, Nara Institute of Science and Technology, 2003.

[133] Katsuya Takanashi, Takehiko Maruyama, Kiyotaka Uchimoto, and Hitoshi Isahara. Identification of "sentences" in spontaneous Japanese detection and modification of clause boundaries ? In *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pp. 183–186, 2003.

[134] Mineki Takechi. The information-ranking business system. *Magazine FUJITSU*, Vol. 51, No. 4, pp. 257–262, July 2000.

[135] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto, and Hozumi Tanaka. Extraction of procedural expressions in a list using surface linguistic cues. In *IPSJ SIG Notes NL-152-2*, pp. 7–14, 2002.

[136] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto, and Hozumi Tanaka. Extracting lists of procedural expressions from web pages. *IPSJ Transaction on Databased (TOD)*, Vol. 44, No. SIG12(TOD19), pp. 51–63, September 2003. in Japanese.

[137] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto, and Hozumi Tanaka. Feature selection in categorizing procedural expressions. In *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages: IRAL2003*, pp. 49–56, July 2003. in Japanese.

[138] Masayuki Takeda, Tomoko Fukuda, Ichiro Naniri, and Mayumi Yamasaki. Discovering characteristic patterns from classical Japanese poem database. *Journal of Information Processing Society of Japan*, Vol. 40, No. 3, pp. 783–795, 1999.

[139] Kazuhiro Takeuchi. *A Study of Text Summarization based on Text Structure Analysis*. PhD thesis, Nara Institute of Science and Technology, 2001.

[140] Kazuhiro Takeuchi and Yuji Matsumoto. An empirical analysis of text structure as a basis for automated text summarization. In *IPSJ SIG Notes NL-133-9*, pp. 61–68, 1999. in Japanese.

[141] Akihiro Tamura, Hiroya Takamura, and Manabu Okumura. Classification of multiple-sentence questions. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pp. 426–437, October 2005.

[142] Ashwin Tengli, Yiming Yang, and Nian Li Ma. Learning table extraction from examples. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, 2004.

[143] Akira Terada, Takenobu Tokunaga, and Hozumi Tanaka. Automatic expansion of abbreviations by using context and character information. *Information Processing & Management*, Vol. 40, No. 1, pp. 31–45, 2004.

[144] Erik Tjong and Kim Sang. Text chunking by system combination. In *Proceedings of Conference on Computational Natural Language Learning*, 2000.

[145] Takenobu Tokunaga. *Information Retrieval and Natural Language Processing*. University of Tokyo Press, 1999. in Japanese.

[146] Noriko Tomuro and Steven L. Lytinen. *New Directions in Question Answering*, chapter Retrieval Models and Q and A Learning With FAQ Files, pp. 183–202. The MIT Press, 2004.

[147] Yuta Tsuboi and Yuji Matsumoto. Authorship identification for heterogeneous documents. In *IPSJ SIG Notes NL-148-3*, pp. 17–24, 2002.

[148] Hiroshi Tsuda, Takanori Ugai, and Misue Kazuo. An approach to automated Web metadata creation for Web directories. In *Proceedings of the 18th Symposium on Informatics*, pp. 17–24, 2002.

[149] Natsuko Tsujimura. *The Handbook of Japanese Linguistics*. Blackwell, 1999.

[150] Hiroaki Tsurumaru, Katsunori Takesita, Katsuki Itami, Toshihide Yanagawa, and Sho Yoshida. An approach to thesaurus construction from Japanese language dictionary. In *IPSJ SIG Notes NL-083*, pp. 121–128, 1991. in Japanese.

[151] Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the ACL 2000*, 2000.

[152] Kanji Uchino, Sachiko Motoi, Minako Hashimoto, Mineki Takechi, Kunio Matsui, and Yasuyo Kikuta. Rule-based text categorization service - a business application of automatic text categorization -. *The Journal of Information Science And Technology Association*, Vol. 50, No. 10, p. 502, 2000.

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

[154] Ellen M. Voorhees. Overview of TREC 2003 Question Answering Track. In *Proceedings of the twelfth Text REtreival Conference(TREC-12)*, 2003.

[155] Ellen M. Voorhees and Donna K. Harman, editors. *TREC:Experiment and Evaluation in Information Retrieval*. The MIT Press, September 2005.

[156] Marilyn Walker, Masayo Iida, and Sharon Cote. Japanese discourse and the process of centering. *Computational Linguistics*, Vol. 20, No. 2, pp. 193–233, 1994.

[157] Joseph Weizenbaum. ELIZA : a computer program for the study of natural language communication between man and machine. *Communication of ACM*, Vol. 9, No. 1, pp. 36–45, 1966.

[158] Terry Winograd. *Understanding Natural Language*. Sangyo Tosyo, Tokyo, Japanese edition, 1976.

[159] Yutaka Yagi, Taiichi Hashimoto, Hideya Mino, Takenobu Tokunaga, and Hozumi Tanaka. On rule ordering in decision lists. In *IPSJ SIG Notes NL-146-4*, pp. 21–26, 2001. in Japanese.

[160] Kazuhide Yamamoto, Shigeru Masuyama, and Shozo Naito. Text summarization by deleting overlapped expressions using related texts. *IEICE Transactions on Information and Systems*, Vol. J79-D-II, No. 11, pp. 1968–1972, 1996.

[161] Yudong Yang and HongJiang Zhang. HTML page analysis based on visual cues. In *Proceedings of 6th International Conference on Document and Analysis*, 2001.

[162] Yang Yiming and Pedersen Jan O. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97 14th International Conference on Machine Learning*, pp. 412–420, 1997.

[163] Ling Yin and Richard Power. Adapting the naive bayes classifier to rank procedural texts. In *Proceedings of ECIR*, pp. 179–190, 2006.

[164] Xinyi Yin and Wee Sun Lee. Understanding the function of web elements for mobile content delivery using random walk models. In *Proceedings of Special interest tracks and posters of the 14th International Conference on World Wide Web*, pp. 1150–1151, 2005.

[165] Minoru Yoshida, Kentaro Torisawa, and Jun'ichi Tsujii. Extracting ontologies from world wide web via HTML tables. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING 2001)*, pp. 332–341, 2001.

[166] Dell Zhang and Wee Sun Lee. Question classification using Support Vector Machines. In *Proceedings of SIGIR-2003*, pp. 26–32, 2003.

[167] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. Fully automatic wrapper generation for search engines. In *Proceedings of the 14th international conference on World Wide Web*, pp. 66–75, 2005.

[168] Ingrid Zukerman and Eric Horvitz. Using machine learning techniques to internet wh-questions. In *Proceedings of ACL-2001*, pp. 547–554, 2001.

[169] 益岡隆志, 田窪行則. 基礎日本語文法. くろしお出版, 1992.

[170] 樺島忠夫. 文章構成法. 講談社, 1980.

[171] 乾健太郎. 文章生成, 第 4 章, pp. 116–153. 電子情報通信学会, 1999.

[172] 鹿島久嗣, 坪井祐太, 工藤拓. 言語処理における識別モデルの発展 –HMM から CRF まで–. 言語処理学会第 12 回年次大会チュートリアル資料, 2006.

[173] 高橋哲朗, 乾健太郎. アノテーションツール”tagrin”の紹介. 言語処理学会第 12 回年次大会論文集, pp. 228–231, March 2006.

[174] 高橋哲朗, 縄田浩三, 乾健太郎, 松本裕治. 質問応答における構文的照合と言い換えの効果. 言語処理学会第 9 回年次大会論文集, 2003.

[175] 篠田義明. コミュニケーション技術　実用的文章の書き方. 中央公論社, 1986.

[176] 手塚芳樹, 橋本泰一, 徳永健伸, 田中穂積. 日本語名詞句のパラフレーズ検索に関する研究. 言語処理学会第 10 回年次大会, pp. 508–511, 2004.

[177] 寺田昭.　自然言語処理を利用したテキストマイニング手法に関する研究. PhD thesis, 東京工業大学, 2003.

[178] 松井くにお. 情報検索における全文検索の高速化及び対話的ナビゲーションに関する基礎的研究. PhD thesis, 東京工業大学, 2003.

[179] 松本吉司, 乾健太郎, 松本裕治. Web ページのテキストセグメント階層構造の抽出. 言語処理学会第 11 回年次大会論文集, 2005.

[180] 森本郁代, 高梨克也, 竹内和広, 小磯花絵, 井佐原均. 話し言葉コーパスへの談話構造タグ付与. 言語処理学会 第 9 回年次大会発表論分集, pp. 695–698, 2003.

[181] 石崎雅人, 伝康晴. 談話と対話, 言語と計算, 第 3 巻. 東京大学出版会, 2001.

[182] 中川裕志, 渡部聰彦. 携帯端末向けコンテンツ変換と自然言語処理. 情報処理, Vol. 43, No. 12, pp. 16–20, 2002.

[183] 長尾確, 加藤恒昭. 談話解析, 第 3 章, pp. 85–107. 電子情報通信学会, 1999.

[184] 麻野間直樹, 古瀬蔵, 片岡良治. 文書構造と言語表現の分析に基づく方法説明抽出. 言語処理学会 第 12 回年次大会論文集, 2006.

[185] 田村晃裕, 高村大也, 奥村学. 質問事項の抽出とその依存関係の特定. 言語処理学会第 12 回年次大会発表論文集, 2006.

[186] 内野寛治, 宗意幸子, 橋本三奈子, 武智峰樹, 松井くにお, 菊田泰代. ルールベースを用いたテキスト分類サービス：自動分類技術のビジネスへの応用. 情報の科学と技術, Vol. 50, No. 10, pp. 497–519, 2000.

[187] 金明哲. 助詞の n-gram モデルに基づいた書き手の識別. 計量国語学, Vol. 23, No. 5, pp. 225–240, 2002.

[188] 木村健司, 徳永健伸, 田中穂積. 日本語名詞句に対する パラフレーズ事例の自動抽出に関する研究. 言語処理学会第 8 回年次大会論文集, 2002.

[189] 野村眞木夫. 『日本語のテクスト—関係・効果・様相—』. ひつじ書房, 2000.

[190] 野田尚史, 益岡隆志, 佐久間まゆみ, 田窪行則. 複文と談話, 日本語の文法, 第 4 巻. 岩波書店, 2002.

[191] 邑本俊亮. 文章理解についての認知心理学的研究. 風間書房, 1998.

Appendix

A Question type definitions

1）Yes–No ：Yes か No での答えを求める質問

-できるか (can) ／できないか (cannot)、あるか／ないか、正しいか／誤りか、など
（例）その氷川丸はまだ航海に使えますか？
（例）ブラウザに保存ボタンはありますか？

2）名称 (Name) ：名称をたずねる質問

-地名や場所は除く
-ものの名前を尋ねるもの (what)
（例）ブラウザとはなんのことですか？

-人の名前をたずねるもの (who)
（例）米国の初代大統領は誰ですか？

-組織の名前を尋ねるもの
（例）世界最大のコンピューターメーカーは？

-Web サイトの名称を尋ねるものを含む

3）叙述 (Description) ：定義／属性／性質／様相／数・量／程度を問うもの

-言葉の意味や事柄の定義を問うもの
（例）「やんつき」の意味を教えてください.
（例）出生地の定義ってなんですか？

-属性、性質、様相を問うもの
（例）肝炎に感染した場合どんな症状が現れますか？

-数・量、程度を問うもの
（例）国内の石油の備蓄量はどのくらいあるのでしょうか？

4）評価 (Evaluation) ：評価／意見をたずねるもの

・質の善し悪しをたずねるもの
（例）A社のデジカメの使い心地はどうですか？
（例）小泉首相の発言をどう思いますか？

・評価／意見を求めるものでも Yes–No を問う形式のものは Yes–No 型に分類する
（例）ワンセグ携帯っていいんでしょうか？

5）方法 (How-to) ：方法を尋ねるもの

（例）エクスプローラをインストールするにはどうすればいいですか？

6）理由 (Reason) ：理由をたずねるもの

（例）なぜＯＳが必要なのですか？

7）場所 (Location) ：場所をたずねるもの

（例）カナダの首都はどこですか？
-URL をたずねるケースを含む

8）時 (Time) ：時や期間をたずねるもの

（例）ノーベル賞の創設はいつですか？

（例）お正月というのは、いつからいつまでのことを言うのですか？

９）相談 (Consultation) ：メインに属する全ての質問タイプのうち、複数の質問タイプが該当しうるが、その特定が難しい質問

-問題や状況（客観）の叙述、したいことなど願望や主観の陳述を行う表現の直前又は直後に、依頼や疑問の表現が続く部分.
（例）今年の夏休みはまとまった休みがとれそうなので、去年の分まで思いっきり楽しみたいです．何か楽しいこと知りませんか？

-依頼や疑問の表現が漠然としていて、複数の質問タイプが想定できるが特定が難しい場合．複数の質問を含んでいても、それが特定できるなら、それぞれの質問タイプを付与する．たとえば次のような例では、異なる質問事項を尋ねている部位には、個々にメインタイプを付与する.
（例）そのお祭りは、いつ、どこでありますか？
-「いつ」には時タイプ、「どこで」には場所タイプを付与する.「いつ、どこでありますか？」を相談タイプとしない.

-質問者自身、質問事項を自覚できていないと考えられる場合

-全ての質問を含みうるような場合
（例）何をどうすればいいのか全然わかりません．教えてください.

１０）その他 (Other) ：他のどのタイプにも属さない質問

B Description type definitions and annotation rules

記述のタイプによって以下の型に分類する。１つの記事に対して複数の型を同時に タグ付けしてよい。いずれのタイプにも属さない場合は、タグを付与しない。

各タイプの定義を下に示す。

１）分析による展開 (Analysis) : まず総論を示し各論の概念を論理的な関係に従って列挙する。次にその列挙した各論を、要素、或いは段階、或いは概念に分解して一つ一つ述べる。階層関係が示される。

２）事実による展開 (Fact) : 事実を淡々と積み重ねていく。これらの事実が総論の内容を支持し、実証し、敷衍して述べていく。

３）実例による展開 (Instance) : タイプ２における事実が、具体例として示されているもの。

４）定義による展開 (Definition) : まず定義を示し、次に定義を明確にするための実例を上げる。定義のみでもよい。

５）時間の順序による展開 (Order of time) : 出来事を起こった順に書く

６）プロセスを記述する展開 (Process) : ものごとの動作、機能のプロセスや手順を記述する

７）結論–理由による展開 (Conclusion-reason) : 最初に結論を述べ、次にその結論を支持する事柄を重要な順に述べる。

８）事象–問題・原因による展開 (Phenomenon-problem) : 何らかの事象や事実を説明したあと、その事象・事実の問題点や原因を説明する。

９）原因–結果による展開 (Cause-result) : 原因と結果から構成される

１０）問題・原因–解決による展開 (Problem-solution) : 初めに問題を述べ、一つ一つの解決法を重要な順又は読み手の関心が高い順に述べる。タイプ９では解決策が示されないがタイプ１０では示される。

１１）比較・対照による展開 (Comparison) : 二つ以上の事柄をくらべながら述べる

各タイプの例文は、文献 [175] を参照のこと。

タグ付け規則１ ：記述タイプの判断に用いた記事中の箇所をタグで囲む。

タグ付け規則２ ：ある記述タイプの説明が、１つの文によって展開されている場合には、その文又はその一部を囲むこともできる。但しその文（又はその一部）が、さらに大きな説明を展開する部分の一部である場合には、大きい方の部分を囲む。このときも、記述タイプの判断に必要な情報のみを囲む。囲む部分の最大長は指定しない。

タグ付け規則３ ：記述タイプが認められる意味的にまとまりのある記事の一部分では、その中の複数の異なる箇所に対して、１つの記述タイプを付与できる。

タグ付け規則４ ：１つの記述タイプで囲まれた部分は交差してはいけない。

タグ付け規則５ ：異なる記述タイプで囲まれた部分は交差してよい。

タグ付け規則６ ：開始タグ及び終了タグは文節の境界にふる。

タグ付け規則７ ：１つの記事のなかに２つ以上の意味的なまとまりがあり、それぞれに説明の展開があるときに、その２つの意味的なまとまりが同じ記述タイプとなるときには、いずれか一方の意味的なまとまりにのみタグを付与する。

List of Publication

Journal Papers

[1] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto and Hozumi Tanaka. Extracting lists of procedural expressions from Web pages. *IPSJ Transaction on Database (TOD)*, Vol. 44, No.SIG12(TOD19), pp.51–63, 2003. in Japanese.

International Conference

[1] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto and Hozumi Tanaka. Feature selection in categorizing procedural expressions. In *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages: IRAL2003*, pp.49–56, Sapporo, July 2003.

Other Publications

[1] Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto and Hozumi Tanaka. Extraction of procedural expressions in a list using surface linguistic cues. In *IPSJ SIG NOTES NL-152-2*, pp. 7–14, 2002. in Japanese.

[2] Mineki Takechi. Information-ranking business system. *Magazine FUJITSU*, Vol. 51, No. 4, pp.257–262, July 2000. in Japanese.

[3] 内野寛治, 宗意幸子, 橋本三奈子, 武智峰樹, 松井くにお, 菊田泰代. ルールベースを用いたテキスト分類サービス：自動分類技術のビジネスへの応用. 情報の科学と技術, Vol. 50, No. 10, pp.502, 2000.

Patents

[1] Mineki Takechi. Information ranking system, information ranking method, and computer-readable recording medium recorded with information ranking program. US Patent No. 834633. Filed: April 16, 2001. Issued: July 19, 2005.

[2] 武智峰樹. テキスト分類プログラム. 特開 2004–348239, 2003 年 5 月 20 日出願.

[3] 武智峰樹. 情報検索プログラム. 特開 2004–157830, 2002 年 11 月 7 日出願.

[4] 武智峰樹. 情報格付けシステム及び情報格付け方法、並びに、情報格付けプログラムを記録したコンピュータ読取可能な記録媒体. 特開 2002–024702, 2000 年 7 月 7 日出願.