A Study on Support-Sentence Retrieval

by

HAI MINH NGUYEN

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Associate Professor Kiyoaki Shirai

School of Information Science
Japan Advanced Institute of Science and Technology

December 2013

To my father, Nguyen Thu Hai
To my mother and to my brother
To my husband

Abstract

In this thesis, we introduce a new task, namely Support-Sentence Retrieval (SSR hereafter), that is centered on sentence retrieval. The goal of a SSR system is to retrieve sentences relevant to a given theme (called support-sentences), then classify them into relevant types such as agreement and contradiction. It would help users to write an article about the theme by giving a comprehensive view of those support-sentences. Our study is the first attempt to develop such a kind of system. The system is divided into two main modules: sentence retrieval and sentence classification.

Sentence retrieval is the task of retrieving relevant sentences against a query. It has been found useful in many other tasks such as question answering, summarization, information extraction, machine translation, etc. Sentence retrieval is usually considered a special case of document retrieval. In fact, the state of the art sentence retrieval method, TF-ISF, is an adaptation of document retrieval method, TF-IDF, at sentence level. However, TF-ISF relies totally on the lexical statistics (term frequency) in the collection. In our system, a full sentence is used as query. Thus, we can utilize the syntactic structure of a sentence in the retrieval. In this task, we propose a method that can utilize both the lexical and grammatical information of a query sentence. In addition, a new query term weighting scheme based on the specificity of the terms is proposed and combined with ordinary IDF weighting for a better performance. Experimental results indicate that our best configuration of sentence retrieval system achieves 32.73% higher precision than the traditional TF-ISF method.

The key idea of sentence retrieval is finding the matching clues (e.g. terms, dependencies) between user's query and candidate sentences. However, extracting lexical information (e.g. query terms, unigram, bi-gram, etc.) for the matching faces the problem of vocabulary mismatch due to the fact that a sentence is very short in comparison with a document.There is very few terms that can be matched. An approach to address this problem is query expansion. Our research resorts to lexical expansion (expanded terms are query-related, e.g. synonym, hypernym). However, this can introduce noise to the system that leads to error in later processes. Therefore, in this thesis, we would also investigate on how to integrate a word sense disambiguation (WSD hereafter) classifier into sentence retrieval system. This is the first attempt to study the impact of WSD in retrieving relevant sentences. We showed that at the moment, due to the limitation of the context information that we can extract from a sentence, a supervised WSD classifier could not predict word sense effectively. In the cases that it is able to identify the correct senses, it is still difficult for the SR system to find the matched terms between query

and candidate sentence although we have already removed the noise added by incorrect expanded terms.

An SSR system extracts sentences relevant to a given topic and put them into meaningful categories, such as agreement and contradiction. Previous researches have already considered the semantic relations between sentences. For example, Recognize Textual Entailments (RTE) task identifies whether the meaning of a text can be inferred from another text; Cross-document Structure Theory (CST) recognizes 18 semantic relations between sentences across topically-related documents. However, most previous approaches applied supervised learning methods which require hand-tagged data. In the sentence classification task of SSR system, we present new sentence classification algorithms based on rules and bootstrapping method to recognize two semantic categories: agreement and contradiction. The initial seed data for training bootstrapping-based classifiers are automatically built. Our best configuration of bootstrapping-based classifiers yields 2.9% higher result than the word overlap baseline in the agreement category. Applying bootstrapping learning even increases the precision at 10 (P@10) of the contradiction class by 12.1% comparing to the rule-based approach. These results are promising due to the fact that the whole process requires no human interaction.

Acknowledgments

There are not many chances in our lives when we have the opportunity to acknowledge the people who really help us to achieve the success and who always encourage us in good and bad situations. I find myself very lucky to have the chance to express my thanks and appreciation to all those kind people in this phD dissertation.

First of all, I would like to thank my principle advisor Associate Professor Kiyoaki Shirai of Japan Advanced Institute of Science and Technology (JAIST). He is the person who has made this work possible by leading it in a feasible direction. Standing behind his advising, I always receive precious comments and constant encouragement which guided me through my most difficult time in research.

I would like to thank Professor Akira Shimazu and Associate Professor Le Minh Nguyen of JAIST for their supportive discussions and suggestions. I am really grateful to Professor Hiroyuki Iida of JAIST for his supervision of my minor research. Be confronted with his challenging questions has furthered my mature in scientific life. I would like to thank Assistant Professor Makoto Nakamura of Nagoya University for his counsel and help during my tough time doing research and living in Japan. I express my gratefulness to Professor Tu Bao Ho of JAIST for his caring from the very first time that I came to Japan. He has helped me quickly get accustom to the unfamiliar environment.

I deeply give my thankfulness to my deceased father, Thu Hai Nguyen, who I admire for his talent and intelligence. He gave me his strength to fight and overcome any problem in life. His kind-hearted behaving towards everyone serves as a shining example for me to follow.

I would like to thank my mother, Lien Chi Nguyen Thanh, who always be positive to face the turbulent life. Her courage and striving in life were the inspiration for me to never give up.

I give the deepest appreciation to my husband, Thanh Duc Chau. It was only his love and empathy that tolerant the strenuous work came to our lives.

I would like to acknowledge my little brother, Nhat Minh Nguyen Hai, who always brings me the peaceful feeling whenever I see him healthy.

I wish to express my gratitude to the Graduate Research Program (GRP) of JAIST that supported me during my 3 years of doctoral study. The program has provided excellent conditions that allowed me to accomplish my research goals successfully.

Last but not least, I wish to thank all of my trustworthy friends for their sharing and encouragements during my everyday life.

I devote my sincere thanks and appreciation to all of them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentence retrieval is a special instance of information retrieval. It has all of the features of an automated information retrieval system, plus its own characteristics. Sentence retrieval has been applied in a variety of natural language processing applications. This chapter introduces a new task called Support-Sentence Retrieval (SSR) and the challenges to develop an effective SSR system.

1.1 Challenge of Sentence Retrieval

Automated information retrieval (IR) systems were originally developed to manage the huge scientific literature since the 1940s with limited users (e.g. researchers and librarians) [31]. Soon later, the target users spread to those of information professionals, such as journalists, lawyers and doctors. With the World Wide Web innovation in recent years, the published information now can reach everyone. Nowadays, information retrieval acts as most people' principle means of information access. IR systems are widely used in many universities, public libraries to provide access to books, journals and other documents. Dictionary and encyclopedia databases also utilize the advantages of the IR systems. In the scope of academic field, Manning *et al.* [69] has defined information retrieval as follow:

> "Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

While there are numerous types of information such as images, videos, audios, etc., most definitions of information retrieval applied to documents [31, 69, 70, 76] because textual information seeking is the most common scenario. In the definition of Manning *et al.*, the information need is expressed as a user query. A query is normally a sequence of terms that describes the user's need. The result outputted by an IR system given a query and a set of documents is a ranked list of documents judged as relevant to the

user's need. This list is sorted in decreasing order by their estimated relevance. The most difficult part of this process is how to understand the query as the exact requirement of the user. In practice, most of the users are reluctant to write more than two or three terms in their queries [116]. The short query therefore may express different meanings which leads to different relevant documents. Therefore, document retrieval is not an easy task for a retrieval system.

Given the fact that documents are formed by a collection of sentences, the sentence retrieval problem that this thesis concerns deals with much shorter unit of information comparing with the document retrieval task. Identifying sentences that are relevant to an input query is an important task in many information retrieval applications. The relevance of sentences may range from expressing the same idea as a query to expanding that idea with additional information. Different criteria are set for different tasks. For example, a question answering system tends to extract relevant sentences containing answers to a user's query [49, 87]. For query-based text summarization, important sentences that have some relationship to the query need to be extracted from the target document [105]. In novelty detection, relevant sentences cannot be redundant [44]. Sentence retrieval is also important in information extraction [92] and machine translation [26]. It is a challenging area that has attracted numerous attention recently [28, 85–87].

Recent researches of Murakami *et al.* develop an application of sentence retrieval called Statement Map [85, 86]. Statement Map aims to help users navigate the numerous amounts of information on the internet and come to informed opinions on their topic of interest. Output of the system is a map of information where view points and evidences of a statement (query) are listed in several semantic classes. Then, the user will judge the reliability of the statement based on this map. This research has attracted us to the field of mining the huge quantity of texts for exploring different points of view.

Let's see a specific example. On September 30, 1993, the Japanese government decided to open its rice market to the world for the first time in three decades due to the sudden drop of harvest amount. Because of a cool, wet summer growing season and an outbreak of blight, Japan faced its worst rice harvest in the post war period – only 74% of the amount of the previous period's harvest – and was forced to purchase, on an emergency basis, foreign rice in the coming 12 months. However, this decision faced many objections because of the fear that foreign rice would be flooding the Japanese market. Obviously a document retrieval system can help us search for all of the relevant information on this issue. Let say a user wants to collect all supporting ideas on the fact that Japan opened its rice market in 1993. A sentence retrieval system is more appropriate for the user to gather the most important clues. Furthermore, a retrieved sentence can be clustered into different categories according to its property. Table 1.1 gives some examples of our ideas to collect support-sentences.

Table 1.1: Example of support-sentences

Query	Rice market in Japan is opened to foreign countries.
Agreement	Japan has been under strong pressure from other member countries of the General Agreement on Tariffs and Trade (GATT), including the United States, to open its rice market.
Agreement	Prime Minister Kiichi Miyazawa's plan to open the Japanese rice market to foreign imports early next year, using "gaiatsu," or pressure from outside, as the excuse for his move, has been stalled by the increasingly uncertain state of the Uruguay Round of multilateral trade talks, political observers say.
Contradiction	The consumer prices of rice are five to seven times those of rice overseas, but the message in the government leaders' remarks is that the planned very high tariff rates would ensure that the nation's rice growers won't face any serious threat from foreign competitors.
Contradiction	Even if the government pledges opening of the nation's rice market to other countries, it might be unable to do so.

Our system, namely Support-Sentence Retrieval (SSR hereafter) provides users all sentences that are relevant to the topic of interest in a categoric way. Our system is similar to the system of Murakami *et al.* [85]. However, their research only focuses on the sentence classification task, which is related to the task of Recognizing Textual Entailment (RTE task [21]) between two text fragments. RTE task has been proposed as a generic task that captures major semantic inferences for many NLP applications such as question answering, information retrieval, information extraction and text summarization.

Besides entailment, there are more semantic relations among text. Cross-document Structure Theory (CST) attempts to characterize 18 kinds of relationships that exist between pairs of sentences coming from one or more documents [105]. In our opinion, semantic relations in CST are too difficult to be distinguished automatically. Furthermore, such detail types of semantic relations are not necessary for most of applications, specifically the SSR system that this study focuses on. A brief explanation of the semantic relations in our SSR system will be discussed in Subsection 1.2.2.

Most of previous studies on analyzing semantic relations between texts applied supervised learning methods to facilitate the semantic features extracted from available annotated corpora [24, 36, 85]. Although the results are good, the cost paid for corpus building is too expensive. The bottleneck of data sparseness is also a serious problem. In this thesis, we study the knowledge based techniques for the task of support-sentence retrieval to avoid this limitation.

1.2 Research Goal

The main goal of this thesis is to develop an effective system that supports a person writing an article on one theme. Given a set of reference documents, people normally find support ideas for a topic by reading all available documents, which is a time-consuming process. It would be better if a system can help us quickly identify which ideas are important and how relevant they are to the topic. The problem is different to a question answering problem because we expect to retrieve broader information, which may provide additional useful information, rather than an exact answer to the query. This information is classified into groups that help the user quickly navigate and capture the main ideas relevant to the being written article. Hence, our final goal does not stop at retrieving sentences relevant to a user query; it goes further to analyze how relevant they are. In this research, this problem is referred to as Support-Sentence Retrieval (SSR). Our study is the first attempt to develop such a kind of system. Two major tasks in the SSR system are sentence retrieval and sentence classification.

1.2.1 Sentence Retrieval

Our first sub-goal is to develop an effective sentence retrieval module of the SSR system, which we called SR module. Although there have been many attention to the task of sentence retrieval, an effective method to solve this problem is still to be found [30]. In the scope of our study, a relevant sentence is the sentence that states the same topic to the input query. It may discuss the same information, the contrast information or extend the topic to other perspectives. Basically, given a set of documents and a query sentence, the sentence retrieval module retrieves the sentences relevant to the query from those documents. For example, given the topic sentence "*Bloods transfusion gives many dangerous viruses*", the system should be able to extract relevant sentences that are both similar and provide additional useful information for users, as the examples 1-3 in Table 1.2. Our research focuses on exploring the effective features of a full-sentence query in a sentence retrieval system. In traditional sentence retrieval systems, a query is usually a collection of words that may not effectively express what a user is looking for. However, humans normally describe ideas in full sentences that contain not only keywords but also semantic relations (dependencies) between the keywords. For example, if only keywords "*blood*," "*transfusion*," "*gives*," "*dangerous*" and "*viruses*" are given as a query, some irrelevant sentences that only include some of the keywords may be retrieved, such as example 4 in Table 1.2.

Our system not only matches keywords but also considers their grammatical and semantic relations. In this study, a full-sentence query is used in a sentence retrieval system for support users writing topic-based articles. Specifically, lexical information and syntactic relations between the query sentence and candidate sentences are used to

Table 1.2: Examples of retrieved sentences for the query "*Bloods transfusion gives many dangerous viruses*"

(1)	Blood transfusion gives man hepatitis E.
(2)	The measures include 2 billion yen to store donated blood for six months within the organization so shipped blood can be quickly recalled if samples are found to have slipped through safety checks, and 4.5 billion yen to remove white blood cells to prevent side effects as well as infectious diseases through blood transfusion.
(3)	An advisory panel for the ministry pointed out at the end of 1997 the necessity of conducting follow-up surveys on blood that may be tainted with hepatitis and other viruses as it was found very difficult to eliminate the danger of infectious diseases completely from blood used for transfusions.
(4)	As viruses have become more dangerous, so integrated security software that enables computer owners to set up firewalls has become mainstream.

calculate their similarity. In addition, the use of a specificity weighting technique based on WordNet's hierarchy to enhance the precision of the sentence retrieval system is proposed.

As we discuss above, using a full-sentence as query may help decrease the ambiguity usually occurrs in a retrieval system that accepts only two or three terms as query. This can be seen as an implicit disambiguation in the system. Regarding applying explicit disambiguating module into an IR system, many previous work have reported positive results [41, 53, 113, 119]. However, to the best of our knowledge, there is no literature review on the impact of Word Sense Disambiguation (WSD hereafter) into sentence retrieval task. As a step to enhance the performance of SR module, we aim to study the effectiveness of applying a WSD classifier to the task of sentence retrieval.

1.2.2 Sentence Classification

Our second sub-goal is to investigate for effective methods to classify the retrieved sentences in SR module into different semantic categories. These methods are empirical test through the classification module (SC module) of the SSR system. The semantic classes should represent different ideas relevant to the query. They act as the advantageous hints for the user's writing. Inspired by the work of Radev *et al.* [105], we consider 5 classes for the SSR system which are agreement, contradiction, subsumption, refinement and cross-reference. Some examples of these semantic classes are presented in Table 1.3. Definition of these 5 classes will be shown in Section 6.3.

In our point of view, the first two relations are the most important ones because they contribute to sketch the main spirit of the article. If the retrieved sentences are all classified in agreement or contradiction category, the article trends to no controversy. On the other hand, if some retrieved sentences are regarded as agreed to the query but some

Table 1.3: Example of 5 semantic classes in SSR system

Query	Sentence	Categories
Smoking is prohibited in almost all public areas.	Smoking inside public and private buildings is strictly prohibited by law.	Agreement, Subsumption
Smoking is prohibited in almost all public areas.	Smoking inside schools prohibited.	Agreement, Refinement
Tap water is safe.	Tap water is generally safe for drinking, though tourists are advised to buy bottled water for drinking.	Subsumption
Tap water is safe.	Generally, tap water is safe to drink after boiling.	Agreement, Subsumption
Tap water is safe.	Tap water is generally not safe for drinking.	Contradiction, Subsumption.
People often visit hot springs to cure their disease.	In Japan, many hot springs in rural locations are maintained by the local government and are open to the public for free, and even expensive spa resort towns usually have at least one public bath open to all for a token fee.	Cross-reference

others are put in contradiction category, there may be disparate opinions on the topic. For this reason, in the scope of this study, we focus on how to distinguish between agreement and contradiction relations between a support-sentence and a query sentence. Our proposed methods to recognize these two categories are developed based on unsupervised features to avoid the bottleneck problem of supervised methods.

1.3 Dissertation Outline

This thesis consists of 7 chapters (see Fig. 1.1). In the next chapter, we discuss on some related tasks and point out our motivation to develop the Support-Sentence Retrieval (SSR) system. The tasks to be discussed include *document retrieval*, *recognize textual entailment* and *cross-document structure theory*.

In Chapter 3, we present an overview of our proposed SSR system. The system is

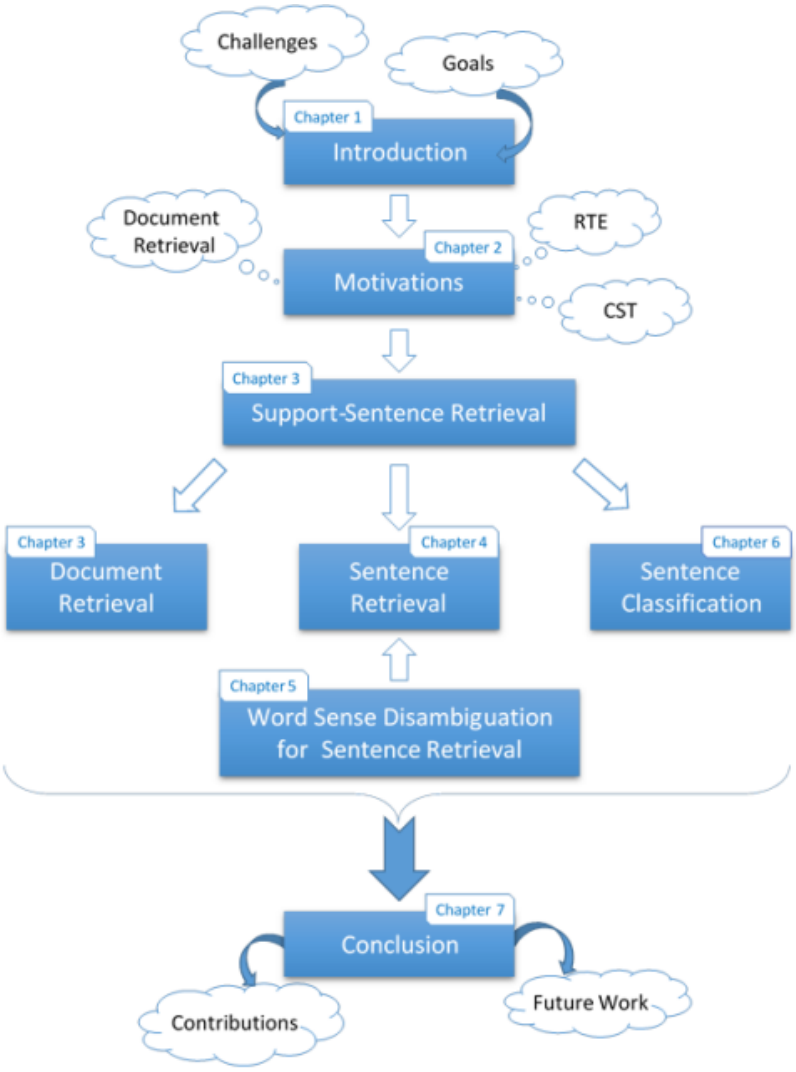

Figure 1.1: Outline of this dissertation

divided into two three main modules: *document retrieval*, *sentence retrieval* and *sentence classification*. In this chapter, we explain the input, output, pre-processes and the first module of the system (document retrieval). The other modules which are two main tasks of our study (sentence retrieval and sentence classification) are explained in the later chapters.

Chapter 4 begins with an overview on our first task: sentence retrieval and a discussion on some previous work in literature. Then, our proposed sentence retrieval method that utilizes both lexical and grammatical characteristics of a sentence are explained in detail. We study the effectiveness of our method in comparison with a standard sentence-retrieval approach. In addition, four kinds of query term weighting schemes are introduced to our proposed retrieval method and are empirically proved to enhance the performance of the sentence retrieval system. Experimental results and some discussion are given to study the effectiveness of our approach.

Chapter 5 presents our study on the impact of applying word sense disambiguation into SSR system. Although using full-sentence query helps decrease the ambiguity of the terms in the query, our proposed term weighting methods in Chapter 4 might introduce noise into the system during the process of query expansion. Therefore, it is necessary to reduce these noise as much as possible through a WSD process. Literature on applying WSD to IR yielded opposite results. Some study said yes, others said no to the enhancement of the IR system when applying WSD. In this chapter, we conduct an experiment to answer the question: whether WSD helps improve the performance of a sentence retrieval system.

In Chapter 6, we study unsupervised methods to recognize two semantic classes: agreement and contradiction. Our proposed rule-based and bootstrapping-based classifiers are constructed using only features extracted from the query and the relevant sentences (output of the SR system in Chapter 4). The initial seeds of our bootstrapping-based classifiers are created automatically. Therefore, the whole process requires no human interaction. This proposal helps exempt the cost of building manually tagged corpus for training the classifiers. The empirical results indicate that our proposed methods give promising results to recognize these two semantic classes.

Finally, chapter 7 will summarize this thesis, point out the contributions and suggest for the future research directions.

Chapter 2

Support-Sentence Retrieval: Motivation

In this chapter, through the discussion of some related tasks to our study, we would like to explain the motivations of our proposed system. Most approaches on sentence retrieval directly adopt methods from document retrieval with some trivial modifications. In Section 2.1, we present a brief introduction of existing *document retrieval* methods and their adoptions in sentence retrieval. Section 2.2 gives some discussions on the problem of applying *query expansion* to sentence retrieval. These tasks motivated us in our first sub-goal. In recent years, bringing semantic relations to sentences is a hot topic. In the final section of this chapter, we review some related tasks (*recognize textual entailment*, *cross-document structure theory*) that lead to our second sub-goal.

2.1 Document Retrieval methods against Sentence Retrieval

The methods on document retrieval models can be classified into three categories as Table 2.1. Although different models are built based on different ideas, all of them have a common fundamental assumption that documents containing more query terms are more relevant to the query [87]. For example, in the *vector space models*, the documents are ranked according to the similarity score of their vector representation and the query vector representation. Each term in the query and the document is weighted by many ways (most are variations of TF-IDF). The common similarity score is the *cosine* of the angle between two vectors. Consequently, documents containing more query terms are regarded as more relevant to the query. In contrast, a sentence usually consists of several singleton query terms. Table 2.2 reveals the average document length (number of terms) and the average sentence length in the corpus of our study[1]. Since sentences are

[1]The corpus is described in details in Subsection 3.1.1.

shorter than documents, a relevant sentence in sentence retrieval task has less chance to contain query terms than a relevant document in document retrieval task. Therefore, the assumption of document retrieval methods does not hold for the case of sentence retrieval. It is clearly that simply applying document retrieval methods to sentence retrieval is not effective. This raises our first motivation to develop a sentence retrieval method that can recognize the discriminating terms in the short context of a sentence.

Alternatively, the *probabilistic models* raised a good idea but never won on performance comparing to other models [69]. It requires some major assumptions (e.g. term independence, terms not in the query don't affect the outcome, etc.). The *language models*, on the other hand, have been proved to significant improve document retrieval performance. However, on other tasks where the unit of retrieval is smaller such as passage retrieval, vector space models are still state-of-the-art models for query-passage scoring [51, 52]. In the task of sentence retrieval, previous researches showed that the vector space model at least performs as well as the best performing empirically tuned and trained sentence retrieval models based on BM25 and language models [29, 67]. These evidences support our decision to choose the vector-space model as a baseline method for the retrieval of support-sentences (Chapter 4).

2.2 Query Expansion and the problem of Sense Ambiguity

Most approaches in sentence retrieval literature are based on regular matching of terms between a query and a sentence. However, due to the fact that a sentence contains a very limited number of terms, not many of these terms can matched the query terms. One method to address this problem is *query expansion*. The work of Losada *et al.* [66] is a comprehensive study on most of the current researches using techniques of query expansion for sentence retrieval. This included well-known term selection techniques which have been used in document retrieval, such as those based on pseudo-relevance feedback [13] and local context analysis [130, 131]. The paper concluded that expansion after sentence retrieval (ASR) with pseudo-relevant feedback and expansion before sentence retrieval (BSR) with local context analysis are the most robust expansion methods for sentence retrieval. However, the cost for expansion ASR is more expensive because we need the result from the initial run of sentence retrieval to collect the feedback. The effect of this kind of feedback is very sensitive to the quality of the initial ranks. Motivated by this, Abdul-Jaleel *et al.* has introduced to use selective feedback, which is more stable but requires a training data [1].

Other studies resort to lexical expansion, i.e. the expansion is assisted by query-related terms (synonyms or related terms from a lexical resource) [137]. The limitation of this type of expansion is that noisy terms can be easily introduced to the new query, mostly

Table 2.1: Existing models for Document Retrieval

Category	Description	Methods
Vector Space Models	Each document is viewed as a vector with each component corresponding to a term in the list of terms. The score of a document d is similarity between the document vector and query vector.	TF-IDF [110] Maximum TF normalization [59] Pivoted document length normalization [117]
Probabilistic Models	Estimate the probability of a term t appearing in a relevant document $P(t\|R = 1)$. This probability helps decide whether documents are relevant or not.	Binary independence model (BIM) [108] Probabilistic logics [32] Inference networks [124] Okapi BM25 [107]
Language Models	Model the idea: words that would likely appear in a relevant document tend to be used in a query. Therefore, a document is a good match to a query if the document model is likely to generate the query. Build a probabilistic language model M_d from each document d, and ranks documents based on the probability of the model generating the query: $P(q\|M_d)$.	Query likelihood [101] Document likelihood [58] Model comparison [55] LM with smoothing [64, 122]

because of the word sense ambiguity. Look at the example in Table 2.3, all sentences contain the word "*arm*", but different context tells different senses of this word. The final column gives the synonyms of the correct sense of the word "*arm*" (taken from WordNet). If we want to expand the word "*arm*" according to WordNet's synonyms, it is necessary to know the correct sense. Otherwise, the expanded terms for this word would lead to errors in later processes. This problem leads to our second motivation of studying how to predict a correct sense of a query term to enhance the effectiveness of query expansion in sentence retrieval (Chapter 5).

Table 2.2: Average length of documents and sentences in Yomiuri corpus

Collection	# terms/sentence	#terms/document
1990	19.4	536.7
1992	19.9	461.7
1993	19.2	427.4
1994	20.3	169.2
1998	20.7	488.5
1999	21.5	514.3
2000	21.7	555.8
2001	21.3	169.5
2002	20.8	546.1
2003	20.6	532.3
Avg.	20.5	440.2

Table 2.3: Example of correct expansion of the word "*arm*" in a sentence

No.	Sentence (target word in bold)	Correct term expansion
1.	Russia is subject to a ban on **arms** exports.	weapon, instrument
2.	The research **arm** of Wako Securities reports an increasing profits.	branch, subdivision, division
3.	The muscles in his **arms** and legs, which do not allow him any mobility, are subject to spasms.	limb

2.3 Consideration of Semantic Relations between Sentences

2.3.1 Recognize Textual Entailments

Many natural language processing applications such as question answering, information retrieval, text summarization need to recognize whether the meaning of a text can be expressed by, or inferred from, another text. Since 2005, the task to capture such semantic relationship between text segments has been introduced and received much attentions. This task is called Recognize Textual Entailments (RTE task) [21]. According to the standard definition, Textual Entailment is defined as a directional relationship between two text fragments, termed Text (T) and Hypothesis (H). It is said that:

T entails H if, typically, a human reading T would infer that H is most likely true.

The traditional RTE Main task which was carried out in the first five RTE challenges from 2005 to 2009 [7, 11, 21, 37, 38] consisted of making entailment judgements over

isolated $T - H$ pairs. This means that the context necessary to judge whether T entails H is only given by T. However, since RTE-6 [10], the RTE Main task had been changed to a Search task, which consists of finding all the sentences in a corpus that entail a given hypothesis. It is noted that in the new Main task, a preliminary information retrieval filtering phase is performed in order to select for each H a subset of candidate entailing sentences to be judged by the RTE systems. Table 2.4 presents a hypothesis referring to a given topic and some of the entailing sentences found in the set of candidate sentences. The examples are taken from the RTE-7 Task Guidelines [9].

Table 2.4: A hypothesis and some of the entailing sentences

	Sentence	Judgement
H	Lance Amstrong is a Tour de France winner	
T_1	Claims by a French newspaper that seven-time Tour de France winner Lance Armstrong had taken EPO were attacked as unsound and unethical by the director of the Canadian laboratory whose tests saw Olympic drug cheat Ben Johnson hit with a lifetime ban.	YES
T_2	L'Equipe on Tuesday carried a front page story headlined "Armstrong's Lie" suggesting the Texan had used the illegal blood booster EPO (erythropoeitin) during his first Tour win in 1999.	YES
T_3	Armstrong, who retired after his seventh yellow jersey victory last month, has always denied ever taking banned substances, and has been on a major defensive since a report by French newspaper L'Equipe last week showed details of doping test results from the Tour de France in 1999.	YES

As a step forward to the binary classification setting of the RTE challenges (entailment or non-entailment), a three-way entailment decision task has been introduced since RTE-3 in 2007 [37]. The three-way task requires each system to decide whether the hypothesis is entailed by the text (ENTAILMENT), incompatible with the text (CONTRADICTION), or neither entailed by nor incompatible with (UNKNOWN). This task is similar to our sub-goal to classify the support-sentences into semantic categories. We roughly summarize the main approaches in RTE challenges in Table 2.5[2]. Among them, *logical inference* is the most intuitive approach. However, in practice, it may be very difficult to formulate a reasonably complete representation of the logical meaning. Recognizing entailment based on *similarity function*, on the other hand, receives a large number of attention and has the diversity in methods, e.g. the simple vector-space models [135], surface string similarity [14], syntatic similarity models [45]. Moreover, the lexical model can be extended with limited context using the most reliable analytic structure (dependency

[2]More details of each approach can be found in [5].

parse). These similarity measures can also be combined together by using a machine learning, e.g. SVM [36, 136]. In this way, RTE task is regarded as a classification problem and can archive good results given good training data [68]. However, the cost paid for building such data is very expensive. This leads to our motivation to develop a rules-based method which utilizes the similarity-function and an unsupervised learning-based methods which can avoid the training data requirement to recognize the semantic classes in our SSR system.

Table 2.5: Existing approaches of RTE task

Approach	Description
Similarity function	compute a similarity score between H and T, then use a threshold to separate RTE classes
Logical form/theorem proving	map the language expressions to logical meaning representations and rely on logical entailment checks by invoking theorem provers.
Machine Learning	extract features from T, H and put to a classifier

2.3.2 Cross-document Structure Theory

Cross-document Structure Theory (CST) is another task of recognizing semantic relations between sentences proposed by Radev [105]. It is used to describe cross-document semantic connections, such as "*elaboration*", "*contradiction*", "*attribution*", "*historical background*", etc. The central idea of CST is to give a set of rhetorical relationships between sentences across topically-related documents. This information can be found useful in multi-document text summarization, semantic entity and relation extraction and non-factoid question answering [138, 139]. A CSTBank corpus of cross-document sentences annotated with CST relations has also been constructed [104]. CSTBank is organized into clusters of topically-related articles. There are 18 CST relationships in total as Fig. 2.1, all of them are domain-independent[3]. Zhang and Radev attempted to classify the CST relations between sentence pairs extracted from topically related documents using weakly supervised machine learning approach [140]. The labelled and unlabelled data used in their study consist of 4,931 and 6,000 sentence pairs, respectively. However, they used a vector space model where the features are class independent and tried multi-class classification. The specific results of only 7 classes were reported and the highest result was of the "*No relationship*" class (88.75%). The other classes had the precision range from 10% to 52.63%. The results may indicate that the recognition methods for each relation should be developed separately [86]. Motivated by this, in this study, we try to develop a semantic classifier that is able to utilize the specific characteristics of each type of relation.

[3]The figure is taken from the paper of Zhang *et al.* [139].

ID	Relationship	Description	Text span 1 (S1)	Text span 2 (S2)
1	Identity	The same text appears in more than one location	Tony Blair was elected for a second term today.	Tony Blair was elected for a second term today.
2	Equivalence (Paraphrase)	Two text spans have the same information content	Derek Bell is experiencing a resurgence in his career.	Derek Bell is having a "comeback year."
3	Translation	Same information content in different languages	Shouts of "Viva la revolucion!" echoed through the night.	The rebels could be heard shouting, "Long live the revolution".
4	Subsumption	S1 contains all information in S2, plus additional information not in S2	With 3 wins this year, Green Bay has the best record in the NFL.	Green Bay has 3 wins this year.
5	Contradiction	Conflicting information	There were 122 people on the downed plane.	126 people were aboard the plane.
6	Historical Background	S1 gives historical context to information in S2	This was the fourth time a member of the Royal Family has gotten divorced.	The Duke of Windsor was divorced from the Duchess of Windsor yesterday.
7	Citation	S1 explicitly cites document S2	An earlier article quoted Prince Albert as saying "I never gamble."	Prince Albert then went on to say, "I never gamble."
8	Modality	S1 presents a qualified version of the information in S2, e.g., using "allegedly"	Sean "Puffy" Combs is reported to own several multimillion dollar estates.	Puffy owns four multimillion dollar homes in the New York area.
9	Attribution	S1 presents an attributed version of information in S2, e.g. using "According to CNN,"	According to a top Bush advisor, the President was alarmed at the news.	The President was alarmed to hear of his daughter's low grades.
10	Summary	S1 summarizes S2.	The Mets won the Title in seven games.	After a grueling first six games, the Mets came from behind tonight to take the Title.
11	Follow-up	S1 presents additional information which has happened since S2	102 casualties have been reported in the earthquake region.	So far, no casualties from the quake have been confirmed.
12	Indirect speech	S1 indirectly quotes something which was directly quoted in S2	Mr. Cuban then gave the crowd his personal guarantee of free Chalupas.	"I'll personally guarantee free Chalupas," Mr. Cuban announced to the crowd.
13	Elaboration (Refinement)	S1 elaborates or provides details of some information given more generally in S2	50% of students are under 25; 20% are between 26 and 30; the rest are over 30.	Most students at the University are under 30.
14	Fulfillment	S1 asserts the occurrence of an event predicted in S2	After traveling to Austria Thursday, Mr. Green returned home to New York.	Mr. Green will go to Austria Thursday.
15	Description	S1 describes an entity mentioned in S2	Greenfield, a retired general and father of two, has declined to comment.	Mr. Greenfield appeared in court yesterday.
16	Reader Profile	S1 and S2 provide similar information written for a different audience.	The Durian, a fruit used in Asian cuisine, has a strong smell.	The dish is usually made with Durian.
17	Change of perspective	The same entity presents a differing opinion or presents a fact in a different light.	Giuliani criticized the Officer's Union as "too demanding" in contract talks.	Giuliani praised the Officer's Union, which provides legal aid and advice to members.
18	Overlap (partial equivalence)	S1 provides facts X and Y while S2 provides facts X and Z; X, Y, and Z should all be non-trivial.	The plane crashed into the 25th floor of the Pirelli building in downtown Milan.	A small tourist plane crashed into the tallest building in Milan.

Figure 2.1: CST relationships and examples

Chapter 3

Proposed System

In this chapter, we present an overview of our proposed system and describe the initial stages of the system.

3.1 System Framework

The Support-Sentence Retrieval (SSR) system receives a sentence as user's query, then find support-sentences in a given document collection. Overview of the system is illustrated in Fig. 3.1. The system includes 4 stages. Firstly, a query and a document collection are given as input. The query used in this study is a sentence rather than a set of keywords as in previous work. In the pre-processing stage, an Indexer will create an inverted index file to store a mapping from each content term in the document collection to its location in the collection. This inverted index file is used in Document Retrieval module in the next stage. The query sentence is also passed through a Query Analyzer to extract features of the query, which are used in the next stages. Three main modules of the SSR system include Document Retrieval, Sentence Retrieval and Sentence Classification. In Document Retrieval module, relevant documents are retrieved by a vector space model with TF-IDF. Next, in Sentence Retrieval module, relevant sentences are extracted from candidate sentences in the set of top 100 retrieved documents. Finally, a semantic label for each relevant sentence is determined in Sentence Classification module. Relevant sentences with their categories are outputted as support-sentences of the query.

In the following subsections, we will explain the input, output of three main modules and the pre-processing steps. Then, we will describe Document Retrieval module in the next section. Details of Sentence Retrieval module will be described in Chapter 4 and 5, while Sentence Classification module is explained in Chapter 6.

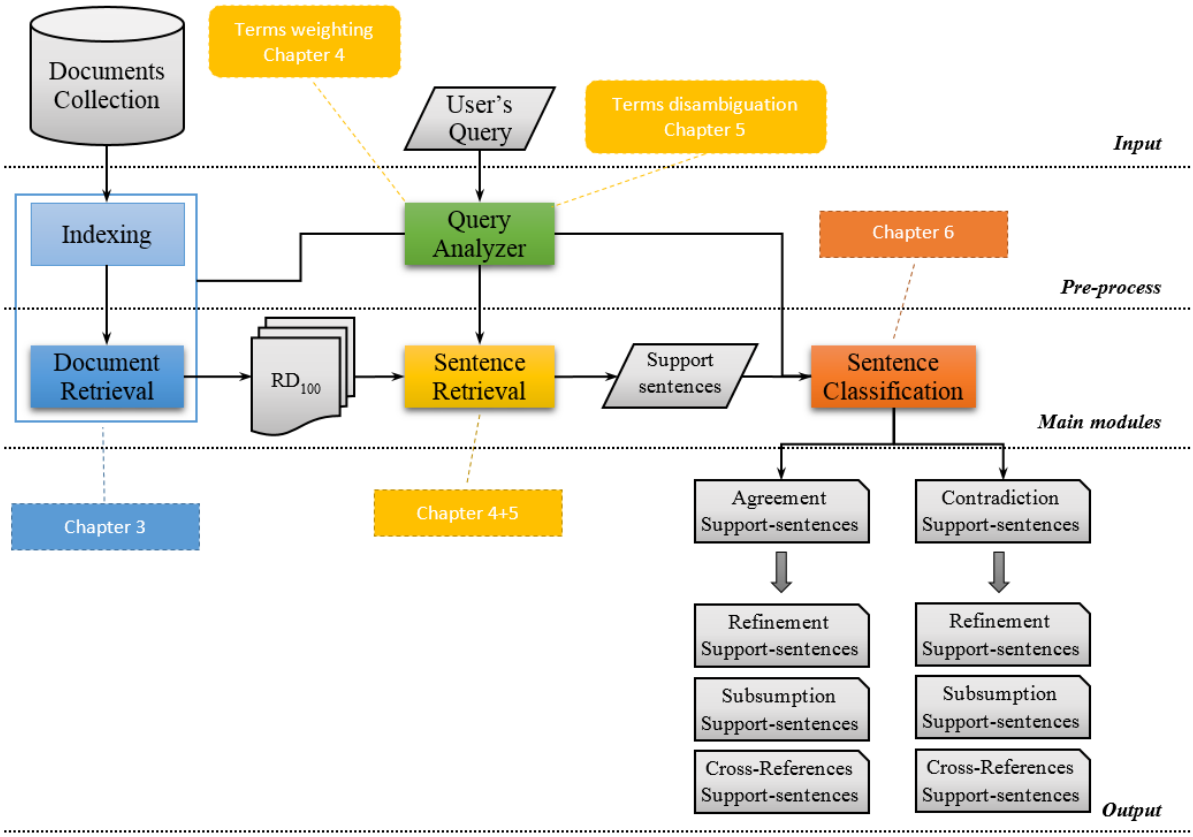

Figure 3.1: SSR system overview

3.1.1 Input

Document Collection

In this study, we use Daily Yomiuri corpus, which contains 10 collections of English news articles from Daily Yomiuri newspaper from 1990 to 2003. Fig. 3.2 shows an example of an article from 1992 collection. The number of articles and sentences of each collection in the Daily Yomiuri corpus is shown in Table 3.1.

User Query

In our study, queries are created by constructing sentences that related to typical events of each year. We looked through the articles and chose several queries for each collection. For example, a query in 1992 collection is: "*Rice market in Japan is opened to foreign countries.*" There are total 55 queries created and tested in the experiments in this research. Table 3.2 shows the number of queries for each collection which are used in Document Retrieval module.

```
<Paragraph ParID="48">
  <Sentence SenID="1">23 .</Sentence>
  <Sentence SenID="2">We confirm the validity of the international debt strategy .</Sentence>
  <Sentence SenID="3">We welcome the enhanced debt relief extended to the poorest countries by
    the Paris Club .</Sentence>
  <Sentence SenID="4">We note that the Paris Club has agreed to consider the stock of debt
    approach , under certain conditions , after a period of three of four years , for the poorest
    countries that are prepared to adjust , and we encourage it to recognize the special situation of
    some highly indebted lower-middle-income countries on a case by case basis .</Sentence>
  <Sentence SenID="5">We attach great importance to the enhanced use of voluntary debt
    conversions , including debt conversions for environmental protection .</Sentence>
</Paragraph>
<Paragraph ParID="49">
  <Sentence SenID="1">Central And Eastern Europe</Sentence>
</Paragraph>
<Paragraph ParID="50">
  <Sentence SenID="1">24 .</Sentence>
  <Sentence SenID="2">We welcome the progress of the democracies in Central and Eastern Europe
    including the Baltic states ( CEECs ) toward political and economic reform and integration into
    the world economy .</Sentence>
  <Sentence SenID="3">The reform must be pursued vigorously .</Sentence>
  <Sentence SenID="4">Great efforts and even sacrifices are still required from their
    people .</Sentence>
  <Sentence SenID="5">They have our continuing support .</Sentence>
</Paragraph>
<Paragraph ParID="51">
  <Sentence SenID="1">25 .</Sentence>
  <Sentence SenID="2">We welcome the substantial mutilateral and bilateral assistance in support of
    reform in the CEECs .</Sentence>
  <Sentence SenID="3">Financing provided by the EBRD is playing a useful role .</Sentence>
  <Sentence SenID="4">Since 1989 , total assistance and commitments , in the form of grants , loans
    and credit guarantees by the Group of 24 and the international financial institutions , amounts to
    $ 52 billion .</Sentence>
  <Sentence SenID="5">We call upon the Group of 24 to continue its coordination activity and to adapt
    it to the requirements of each reforming country .</Sentence>
  <Sentence SenID="6">We reaffirm our readiness to make fair contributions .</Sentence>
</Paragraph>
```

Figure 3.2: Example of a document from the Daily Yomiuri corpus

Table 3.1: Statistics from the Daily Yomiuri corpus

Collection	#Articles	#Sentences
1990	3,961	109,677
1992	7,616	176,600
1993	11,570	258,009
1994	13,296	262,211
1998	9,676	228,096
1999	9,800	234,576
2000	9,082	232,997
2001	8,660	229,579
2002	8,828	231,249
2003	8,677	224,235
Total	91,166	2,187,229

Table 3.2: Number of queries for each collection

Collection	1990	1992	1993	1994	1998	1999	2000	2001	2002	2003	Total
# Queries	6	5	5	5	7	4	6	4	5	8	55

3.1.2 Output

Our system consists of three main modules; the output of each module will become an input of the next module (see Fig. 3.3). Each module's output is as follows:

Document Retrieval: the documents that are relevant to the input query, sorted by scores of relevance (set of relevant documents in Fig. 3.3). Details of the algorithm used in Document Retrieval module is discussed in the next section.

Sentence Retrieval: the sentences that are relevant to the input query, sorted by relevance scores (set of relevant sentences in Fig. 3.3). These sentences are called support-sentences and the definition of them is given in Chapter 4.

Sentence Classification: the semantic classes of the previous retrieved support-sentences (e.g. sets of Agreement and Contradiction in Fig. 3.3). We introduce 5 semantic classes in Chapter 6. This is also the final output of our system.

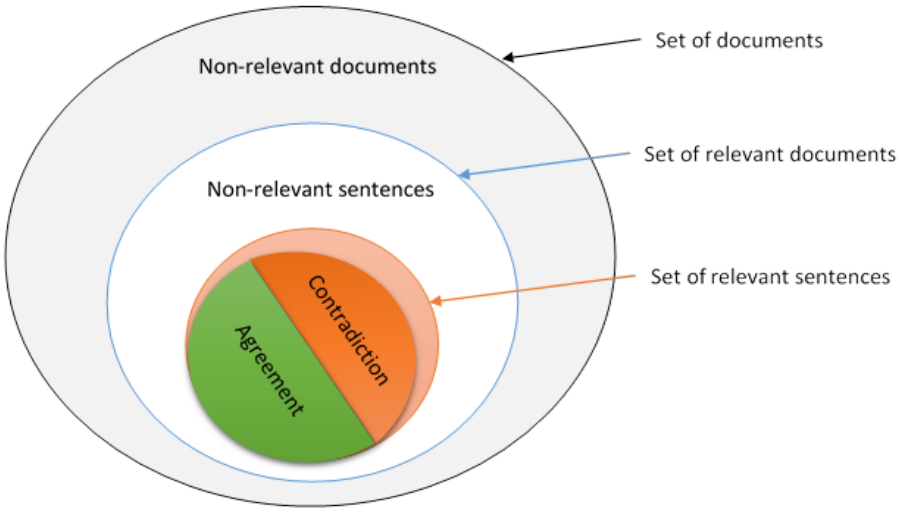

Figure 3.3: The sets of relevant documents, non-relevant documents, non-relevant sentences and support-sentences

3.1.3 Pre-process

Indexing

In the indexing process, all terms in all documents are examined to generate a list of indexed terms. This list is used to build an inverted index, one of the most commonly used file structure for information retrieval. The structure of an inverted index is illustrated in Fig. 3.4[1]. Each entry of the inverted file contains a keyword (indexed term) and the document ID which is a unique identifier for a document containing that keyword. The retrieval process is done by looking up query terms in the inverted file.

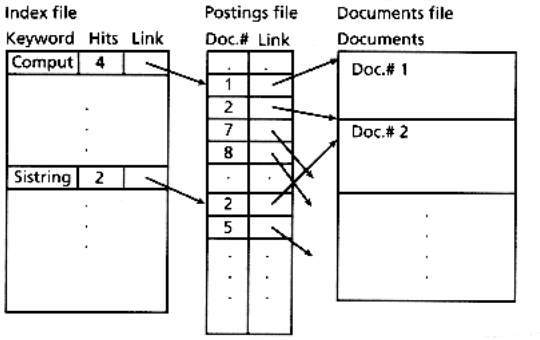

Figure 3.4: An inverted file structure

The automatic indexer in our system is implemented as follows:
(1) Split all words in all documents and convert them to lower case.
(2) Stem all words using Porter stemming algorithm [102].
(3) Remove the words that appear in the stopwords list[2].
(5) Sort all terms by alphabetical orders.
(6) Link each term with its corresponding document.

Query Analyzer

In Query Analyzer, the input query is split into words and stemmed using Porter stemming algorithm. Different modules of the system require further process of the query. For example, the query sentence is syntactically parsed, query terms are added weights in Chapter 4 and disambiguated in Chapter 5. We will discuss the Query Analyzer for each module in the corresponding chapters.

[1]The figure is taken from [31].
[2]The stopwords used in our study is listed in Appendix A.

3.2 Document Retrieval

The first module of SSR system is Document Retrieval (DR), in which we apply a vector space model with TF-IDF weighting for the retrieval of relevant documents. This model has been proved to be robust in document retrieval. TF-IDF stands for *term frequency* (TF) and *inverse document frequency* (IDF). Term frequency denotes the number of time a term t occurs in document d:

$$\mathrm{TF}_{t,d} = n_{t,d} \tag{3.1}$$

Since it is clearly that all words in a document are not equally important [69, pp. 108], inverse document frequency is used to attenuate the effect of terms that occur too often in the document to be meaningful for the relevance determination. For instance, a collection of documents on the economic topic is likely to have the term "*finance*" in almost every document. Such terms may have little or no discriminating power in determining the relevance and should have a low weight. IDF of term t in document d is computed as follows:

$$\mathrm{IDF}_t = \log \frac{N}{\mathrm{DF}_t} \tag{3.2}$$

where N denotes the total number of documents in the collection, DF_t is the document frequency of the term t (the number of documents in the collection that contain term t). The idea of IDF is that the more common a term is, the lower its weight. So, IDF value of a rare term is high, whereas its value of a frequent term is low.

Finally, term frequency and inverse document frequency are combined to produce a composite weight for each term t in the document d by the following formula:

$$\text{TF-IDF}_{t,d} = \mathrm{TF}_{t,d} \times \mathrm{IDF}_t \tag{3.3}$$

The underline meaning of the TF-IDF weight is to assign the highest value to the term that occurs a lot within a small number of documents (which is the case that this term has the discriminating power over other terms). For those terms that occur just a few time in a document or in many documents, the TF-IDF for them are lower. If a term occurs in every document, then the TF-IDF of it is the lowest.

At this point, each document d in the corpus is represented by a vector $\overrightarrow{V}(d)$ with one component corresponding to each term in the inverted index, each component has a weight that is given by Eq. (3.3). For the terms that do not appear in a document, this weight is 0. The query q is also converted into a vector ($\overrightarrow{V}(q)$) by the same way. Then, the similarity (or the relevance score) between d and q is quantified by the cosine similarity of their vector representations.

$$\mathrm{sim}(d,q) = \cos\left(\overrightarrow{V}(d), \overrightarrow{V}(q)\right) = \frac{\overrightarrow{V}(d) \cdot \overrightarrow{V}(q)}{|\overrightarrow{V}(d)||\overrightarrow{V}(q)|} \tag{3.4}$$

Finally, all documents are sorted by this similarity score. Top K documents are regarded as relevant documents. In our experiment, K is set to 100. These 100 top ranked documents are given as input for the next module: Sentence Retrieval, which is discussed in the next chapter.

Chapter 4

Retrieval of Support-Sentences

The main content of this chapter is the proposal of a Sentence Retrieval (SR) module for our SSR system. Firstly, we introduce the sentence retrieval task and its related work. Then, the proposed SR system is explained and experimented. Discussion is given out at the end of this chapter.

4.1 Overview

Given a set of documents C_D that are relevant to a query Q, sentence retrieval task consists of finding sentences in C_D that are relevant to Q. This task has attracted numerous attention recently due to its usefulness in a wide range of Information Retrieval applications, such as summarization, novelty detection, question answering and opinion mining [4, 28, 65, 67, 87]. The viewpoints of relevance are different according to specific applications. For example, in novelty detection, the expected output is the set of non-redundant sentences that are not only similar to the query but also provide new information (novel sentences) [62]. In our study, a relevant sentence is called support-sentence and is defined as follows.

A support-sentence S is a sentence relevant to an input query Q by the following criteria:

- S is on the same topic with Q
- S can be agreed or conflict with Q
- S can introduce more information, discussion referent to the topic in Q.

Examples of support-sentences for a query are given in Table 4.1. All of them are regarded as relevant to the topic of the query because they all discuss on the contribution of Japan in United Nation peacekeeping operations. However, some of them say that Japan has not contributed enough for the peacekeeping operations (sentence 1, 2) while other sentences give some activities to show that Japan has done its mission well (sentence 3, 4, 5).

Table 4.1: Example of support-sentences for the query "*Japan greatly contributes U.N. peacekeeping operations.*"

No.	Sentence
1	Komeito, for instance, is insisting that the bill eliminate the possible dispatch of Japanese personnel for U.N. peacekeeping operations.
2	Some U.N. observers say that Japan, which has expressed its desire to become a permanent member of the Security Council, should play a more active role in peacekeeping operations.
3	In other words, Japan has declared its determination to tackle in good faith some aspects of U.N. peacekeeping operations, such as monitoring of ceasefires and elections-a position supported by a large segment of the Japanese public.
4	The government thus plans to send several hundred SDP personnel to Cambodia to provide such logistical support, and this will mark the start of Japan's full-scale participation in U.N., peacekeeping missions and the start of the fulfillment of its pledge to contribute more to the world.
5	While Japan has previously sent only 33 people to monitor elections as part of U.N. peacekeeping operations in Namibia and Nicaragua in past years, the approved plan calls for sending more than 1,800 SDF personnel to Cambodia under the auspices of the U.N. Transitional Authority in Cambodia.

In short, the main contributions of this chapter include:

- Proposal of a hybrid approach for retrieving support-sentences. This approach takes into account both lexical and syntactic information of a sentence in the query matching process.
- Proposal of new weighting schemes that is able to capture the specificity of each term in the query.
- Suggestion of two evaluation criteria that are fit for support-sentence retrieval.
- Showing effectiveness of the proposed method by two experiments.

In the next section, we review some previous work on sentence retrieval. Section 4.3 includes our proposed system for the SR module. Section 4.4 and 4.5 present the experiments set up and empirical results along with some discussions. Finally, Section 4.6 will summary this chapter.

4.2 Related Work

Sentence retrieval is usually regarded as a special case of document retrieval. Hence, most researches in sentence retrieval area proposed methods that acquired from document retrieval models (such as TF-IDF, BM25, query likelihood, etc.) [4, 28, 87]. The central idea of these models is trying to estimate a match score between query terms and sentence terms. However, as Murdock indicated in her research, sentence retrieval differs from document retrieval in many ways [87]. Because a sentence contains a short piece of information compared to a whole document, the problem of vocabulary mismatch is much more serious in sentence retrieval. Therefore, simply applying document retrieval techniques to sentence retrieval is ineffective. The general principle of sentence retrieval is to extract useful information from a query and search for this information in the given documents. A query expansion technique is usually applied to enrich the query due to its shortage of information. Expansion based on pseudo-relevant feedback [131] is a very common method. However, it inappropriately works when the query contains ambiguous terms. Other researches use lexical resources (e.g., WordNet) to find the expanded terms [8, 126]. Nevertheless, the results are not as good as expected, mostly because of the difficulty of word sense disambiguation in short queries.

In our study, a query is a full sentence which is able to carry richer information than a set of keywords. Moreover, the context of a sentence usually helps to specify the meanings of the ambiguous words without disambiguation. To this end, our idea of finding support-sentences is related to finding similar questions in community-based question answering (cQA) services [46, 83, 128], except that question-to-question matching in cQA is much stricter than query-to-support-sentence matching in our system. Traditional question retrieval approaches use the *vector space model* [47, 48]. However, Wang *et al.* indicated that exploiting the syntactic structure is more effective in capturing the similarities between questions [128]. In spite of that, applying only syntactic matching may not be effective for our system as Cui *et al.* proved that using a strict match of syntactic structure is problematic due to the sparse data [20]. Therefore, in this study, we take both the query's syntactic and lexical information into account. In addition, query terms are enriched using a lexical resource [81]. In this way, we are able to obtain additional information in retrieved sentences rather than only information that has been stated in the query.

The advantage of using syntactic features in previous work was to consider term dependencies in a query for sentence retrieval. However, previous studies tend not to consider term weighting in syntactic relations [20, 97, 98]. Wang *et al.* gave higher priority to verbs and nouns in a query by boosting their weight [128]. In our study, we extend this idea by giving not only the verbs and nouns but also all content words in the query appropriate weights based on their importance. Recently, Losada *et al.* proposed a term weighting approach that utilizes statistical information of query term in the collection [67]. In

this approach, a high term frequency (HTF) score is given to each significant word in a sentence. The similarity score between a sentence and a given query is the sum of the common term frequency-inverse sentence frequency (TF-ISF) and HTF scores. The authors indicated that sentences with poor overlap can be retrieved if they contain highly frequent terms. However, if the highly frequent terms are common words (but still are content words), this approach might not be appropriate because general words may not provide specific information that satisfies the user's need. Such words are usually specialized terms that appear a lot in a certain domain. For example, in ecology documents about animal habitats, the word "*animal*" may appear a lot. When a user wants to find sentences related to the habitat of a bat, the sentences containing the word "*animal*" will have a high score but may not contain the requested information. Hence, the correctness of this method depends on the specificity of the high frequency word. In our work, we propose a weighting scheme that generalizes Losada's idea. In our approach, each term in the query is given a score with regards to its specificity level from very specific to very general. Our hypothesis is that the more specific the word, the more details it provides (so it is important). Therefore, sentences that contain such kinds of words may better satisfy user's need.

4.3 Sentence Retrieval: Proposed System

Fig. 4.1 shows an overview of our SR module. Firstly, relevant documents from the DR module (see 3.2) and the user's query are input to the system. Then, Query Analyzer will splits the query into terms and give each term a weight value. Each candidate sentence (taken from the relevant documents) is parsed and matched to the query by our Hybrid matching algorithm. If the matching score is lower than a threshold T, the candidate sentence is regarded as non-relevant; otherwise, it is put to the set of relevant sentences. Finally, all relevant sentences are sorted by their matching scores. If two sentences have the same scores, the longer sentence is in higher order. The ranked list of relevant sentences is the output of SR system. In the next part, we present our Hybrid approach to retrieve relevant sentences. The query term weighting techniques are described in Subsection 4.3.2.

4.3.1 Hybrid Approach for Sentence Retrieval

Let us suppose Q is a query sentence, S is a candidate sentence in relevant documents. For each S, we calculate the similarity between Q and S, SimScore(Q,S) in Eq. (4.1). If SimScore(Q,S) is greater than a threshold T, S is retrieved as relevant sentence.

$$\text{SimScore}(Q, S) = \beta \cdot \text{SynScore(Q,S)} + (1 - \beta) \cdot \text{LexScore(Q,S)} \tag{4.1}$$

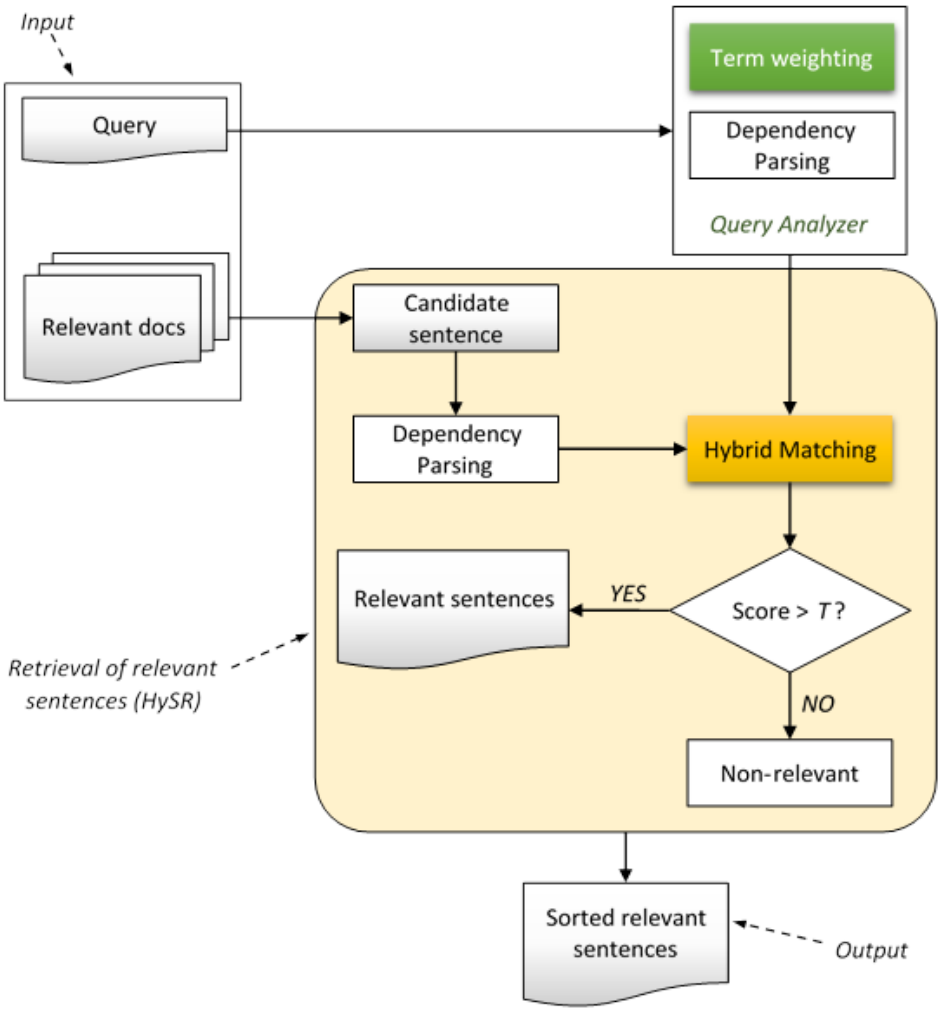

Figure 4.1: Structure of the Sentence Retrieval system

where β is an adjustment parameter which represents the trade-off between SynScore(Q,S) and LexScore(Q,S), T is set to 0 in the experiments in Section 4.4.

As shown in Eq. (4.1), we consider both syntactic score SynScore(Q,S) and lexical score LexScore(Q,S). Since a query is a full sentence in our system, we can consider the syntactic dependencies between content words in the query. If two sentences share the same or similar dependencies, they are likely to be relevant. However, the sentence structure can be transformed without changing the meaning (for example, by changing the active-passive structure). In order to capture relevance between sentences in different syntactic structures, similarity between words, i.e. LexScore(Q,S), is also considered. We call our method *Hybrid Sentence Retrieval* (**HySR**).

Syntactic Score

In this study, we use Stanford Parser [72] to obtain the typed dependencies of a sentence. Each dependency provides a simple description of the grammatical relationship between

two words in the sentence. Fig. 4.2 gives an example of the output from Stanford Parser for the query "*Visitors like to visit Japan in cherry blossom season.*"[1] The output includes part-of-speech tags, a constituent tree and dependency representations (basic and collapsed). As shown in Fig. 4.3, these dependencies map straightforwardly onto a directed graph representation, in which words in the sentence are nodes in the graph and grammatical relations are edge labels[2]. After we acquire the dependencies of both Q and S, a syntactic score between Q and S is computed as follows:

$$\text{SynScore}(Q, S) = \frac{1}{2N_d} \sum_i \sum_j DS(dep_j, dep_i) \tag{4.2}$$

where N_d is the number of pair of dependencies between Q and S, DS is a matching score between each dependencies in Q (e.g. dep_i) and each dependencies in S (e.g. dep_j). DS is calculated as follows:

$$DS(rel_j(s_1, s_2), rel_i(q_1, q_2)) = \begin{cases} 0 & \text{if } rel_i \neq rel_j \\ SS(s_1, q_1) + SS(s_2, q_2) & \text{otherwise} \end{cases} \tag{4.3}$$

In Eq. (4.3), $SS(s, q)$ is a binary function indicating whether the word s belongs to the set of synonyms of the word q in WordNet. That is, for each typed dependency $rel_i(q_1, q_2)$ in the query, we try to expand q_1 and q_2 with their synonyms. In this way, we loosen the strict matching criteria that was used in previous work. Let $Syn(q)$ be the set of synsets of the word q, $SS(s, q)$ is computed as in Eq. (4.4).

$$SS(s, q) = \begin{cases} 0 & \text{if } s \notin Syn(q) \\ 1 & \text{otherwise} \end{cases} \tag{4.4}$$

Note that in Eq. (4.4), the sum of $DS(dep_i, dep_j)$ is divided by $2N_d$ to normalize the score to [0,1].

Lexical Score

LexScore(Q,S) evaluates similarity between all pairs of content words in two sentences. If two contents words are identical or similar, we increment the lexical score. Let Q_c and S_c be a set of content words in Q and S, respectively. LexScore(Q,S) is defined as follows:

$$\text{LexScore}(Q, S) = \frac{1}{|Q_c||S_c|} \sum_{q \in Q_c} \sum_{s \in S_c} SSH(q, s) \tag{4.5}$$

[1]The example is taken from Stanford Parser online demo: http://nlp.stanford.edu:8080/parser/index.jsp

[2]Definitions of 53 Stanford typed dependencies can be found in [23].

Your query

Visistors like to visit Japan in cherry blossom season.

Tagging

```
Visistors/NNS  like/VBP  to/TO  visit/VB  Japan/NNP  in/IN  cherry/JJ  blossom/NN  season/NN  ./.
```

Parse

```
(ROOT
  (S
    (NP (NNS Visistors))
    (VP (VBP like)
      (S
        (VP (TO to)
          (VP (VB visit)
            (NP
              (NP (NNP Japan))
              (PP (IN in)
                (NP (JJ cherry) (NN blossom) (NN season))))))))
    (. .)))
```

Typed dependencies

```
nsubj(like-2, Visistors-1)
root(ROOT-0, like-2)
aux(visit-4, to-3)
xcomp(like-2, visit-4)
dobj(visit-4, Japan-5)
prep(Japan-5, in-6)
amod(season-9, cherry-7)
nn(season-9, blossom-8)
pobj(in-6, season-9)
```

Typed dependencies, collapsed

```
nsubj(like-2, Visistors-1)
xsubj(visit-4, Visistors-1)
root(ROOT-0, like-2)
aux(visit-4, to-3)
xcomp(like-2, visit-4)
dobj(visit-4, Japan-5)
amod(season-9, cherry-7)
nn(season-9, blossom-8)
prep_in(Japan-5, season-9)
```

Figure 4.2: Example ouput of Stanford Parser for the query "*Visitors like to visit Japan in cherry blossom season.*"

$SSH(q, s)$ is 1 if s is a synonym or hypernym of q in WordNet, otherwise 0. Note that LexScore(Q,S) is normalized to [0,1].

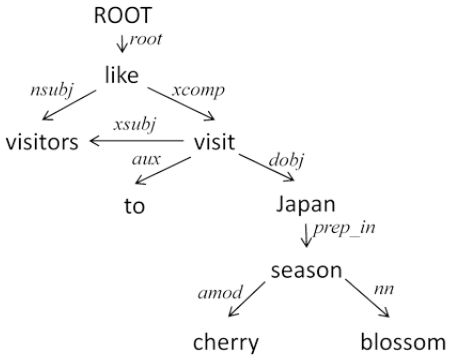

Figure 4.3: Graphical representation of the Stanford Dependencies for the sentence:"*Visitors like to visit Japan in cherry blossom season.*"

4.3.2 Query term weighting for Sentence Retrieval

In order to recognize the important terms and their dependencies, we also employ query weighting in the system. In this section, we describe four weighting schemes: equal weighting, IDF weighting, specificity weighting and combining weighting. Let q_i be the word at position i in the query sentence Q. Our system gives each q_i a weight w_{q_i} as in Eq. (4.6), where $f(q_i)$ is a weighting function altered by different weighting schemes described later. w_{q_i} is added into the SR system by modified DS as in Eq. (4.7) and LexScore as in Eq. (4.8) for each term q_i.

$$w_{q_i} = \begin{cases} 0 & \text{if } q_i \text{ is a stop word} \\ f(q_i) & \text{otherwise} \end{cases} \tag{4.6}$$

$$DS(rel_j(s_1, s_2), rel_i(q_1, q_2)) = \begin{cases} 0 & \text{if } rel_i \neq rel_j \\ w_{q_1} \times SS(s_1, q_1) + w_{q_2} \times SS(s_2, q_2) & \text{otherwise} \end{cases} \tag{4.7}$$

$$\text{LexScore}(Q, S) = \frac{1}{|Q_c||S_c|} \sum_{q_i \in Q_c} \sum_{s_j \in S_c} w_{q_i} \times SSH(q_i, s_j) \tag{4.8}$$

Equal Weighting

In the simplest case, $f(q_i) = \text{EQUAL}(q_i) = 1$, which means all terms have equal weights. As we explained in Section 4.1, this equal weighting cannot help indicating important words in the query.

IDF weighting

An IDF weight implies whether a term is common or rare across all documents. The more common a term, the lower its IDF value. Thus, IDF weighting enables us to alleviate the dominant words that are popular but still content words. These words are usually general

terms that appear a lot in a document set but do not provide detail information. The weighting function for IDF weighting is defined as follows:

$$f(q_i) = \frac{1}{N_{idf}} \log \frac{|D|}{df(q_i) + 1} \tag{4.9}$$

where $|D|$ is the total number of documents in a corpus and $df(q_i)$ is the number of documents containing the term q_i. The ordinary IDF score is divided by N_{idf} for normalization, where N_{idf} is the maximum IDF score ($N_{idf} = \log |D|$).

Specificity weighting

Intuitively, it can be observed that the more specific a word in the query, the more important (detail) it plays in a user's need of information. In the query sentence "*Visitors like to visit Japan in cherry blossom season*," the most important hints in the user's query are "*cherry*" and "*blossom.*" These words are more specific than the other words. These implicit hints can be found easily by looking in the hierarchical representation of words in WordNet, in which "*cherry*" and "*blossom*" are located at deeper positions than other words such as "*season*". That is, the WordNet hierarchy can be used to infer the specificity of a word. We will give more weight to specific words such as "*cherry*" and "*blossom*" than other words. To infer the specificity of a word, we suppose the following hypotheses:

> **Hypothesis 1:**
> Let l be the deepest leaf node of the hyponym tree of the word q and $h(q, l)$ be the height from q to l. The smaller $h(q, l)$ is, the more specific q is.

> **Hypothesis 2:**
> Let e be the highest node (root node) of the hypernym tree of the word q and $h(e, q)$ be the height from e to q. The larger $h(e, q)$ is, the more specific q is.

The weighting function for specificity weighting of a query term q_i, SPEC(q_i), is computed as follows:

$$f(q_i) = \text{SPEC}(q_i) = \frac{h(e_i, q_i) + (\alpha - h(q_i, l_i)) + 1}{2\alpha + 1} \tag{4.10}$$

where α is the maximum height of $h(e_i, q_i)$ and $h(q_i, l_i)$ and is set to 15 in this study[3]. When q_i has more than one sense, $f(q_i)$ is the average of SPEC(q_i) for all senses. Table 4.2 shows the hyponym tree of the word "*cherry.*" The deepest leaf node of this tree is the node "*blackheart, blackheart cherry*"; hence, $h(\text{"}cherry\text{"}, l) = 3$. Table 4.3 shows the hypernym tree of "*cherry*". The height from the "*cherry*" node to the "*entity*" node is 11; hence, $h(e, \text{"}cherry\text{"}) = 11$. Therefore, SPEC("*cherry*")= 0.77.

[3] α is set to 15 according to the maximum height of words in WordNet.

Table 4.2: Hyponym tree of *cherry.noun*

cherry
⇒ sweet cherry, black cherry
⇒ bing cherry
⇒ heart cherry, oxheart, oxheart cherry
⇒ blackheart, blackheart cherry
⇒ capulin, Mexican black cherry
⇒ sour cherry
⇒ amarelle
⇒ morello

Table 4.3: Hypernym tree of *cherry.noun*

cherry, cherry tree
⇒ fruit tree
⇒ angiospermous tree, flowering tree
⇒ tree
⇒ woody plant, ligneous plant
⇒ vascular plant, tracheophyte
⇒ plant, flora, plant life
⇒ organism, being
⇒ living thing, animate thing
⇒ object, physical object
⇒ physical entity
⇒ entity

Combining weighting

Although specificity weighting is good for elevating the importance of specific terms in a query, it is only appropriate for nouns and verbs because WordNet only provides hypernyms and hyponyms for these two parts of speech. Therefore, IDF weighting may be useful as a backup when the specificity score cannot be computed using WordNet. In order to combine the advantages of these two weighting schemes, we use a function as in Eq. (4.11) (Note that these two weighting values are scaled to $[0, 1]$.)

$$f(q_i) = \mathrm{COMB}(q_i) = \begin{cases} \mathrm{SPEC}(q_i) & \text{if } q_i \text{ is in WordNet} \\ \mathrm{IDF}(q_i) & \text{otherwise} \end{cases} \tag{4.11}$$

4.4 Evaluation Configuration

4.4.1 Experiments Setup

Document collection described in Subsection 3.1.1 is used in the experiments. It consists of 10 collections of Daily Yomiuri newspaper from 1990 to 2003. As described in 3.1.1, we prepared several queries for a collection of each year, 55 queries in total. The results of different sentence retrieval systems for these queries are evaluated and compared for each collection of a year as well as the whole document collection.

In order to evaluate our proposed methods, we set up two experiments as follows:

Experiment A: HySR

In this experiment, we evaluate the proposed HySR algorithm as described in Subsection 4.3.1. The baseline used in comparison is the TF-ISF model. For testing the proposed HySR algorithm, we evaluate different values of β: $\beta \in \{0, 0.2, 0.5, 0.7, 1\}$.

Experiment B: HySR+Weights

In this experiment, we evaluate the effectiveness of the support-sentence retrieval system that exploits the HySR algorithm together with different query terms weighting schemes. We study four different weighting schemes as described in Section 4.3.2: EQUAL, IDF, SPEC and COM.

4.4.2 Baseline

As we discussed in Section 2.1, there has been evidence from previous research that the vector space model is a strong baseline for sentence retrieval task. Fernandez has conducted experiments to provide evidence that TF-ISF (a vector space model for sentence retrieval adapted from TF-IDF) is a very competitive baseline [28]. In his study, the comparison is performed among TF-ISF, Okapi BM25 [109] and a Language Modeling approach based on the Kullback-Leibler divergence (KLD) [57]. Results showed that TF-ISF performed at least as well as tuned BM25 and both methods outperformed KLD significantly. These results demonstrated that TF-ISF is an effective method that performs similarly to an optimal BM25 model. In addition, TF-ISF is parameter-free, which is another advantage. Thus, in this study, we also use this simple vector space model as the baseline.

In this model, both the query and candidate sentences are represented as weighted vectors. The candidate sentence is scored based on its similarity to the query vector. This sentence weighting function is a variant of TF-IDF in document retrieval. Note that stop words and functional words are not considered in vector building. In this manner, the

query is just a collection of terms with no grammatical relations. The relevance score of sentence given query is computed as in Eq. (4.12).

$$\text{SimScore}_{tfisf} = \sum_{t \in Q} \log(tf_{t,Q} + 1) \cdot \log(tf_{t,S} + 1) \cdot \log\left(\frac{n+1}{0.5 + sf_t}\right) \tag{4.12}$$

where $tf_{t,Q}$ and $tf_{t,S}$ are the occurrences of term t in query Q and sentence S, respectively; sf_t is the number of sentences that contain term t; and n is the number of sentences in the collection.

4.4.3 Measurements

The two most frequent and basic measures for information retrieval effectiveness are precision and recall [69, pp. 142]:

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}) \tag{4.13}$$

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant}) \tag{4.14}$$

A trade off measure for precision and recall is the F measure, which is the weighted harmonic mean of precision and recall:

$$F = \frac{(\beta^2 + 1)\text{Precision} \times \text{Recall}}{\beta^2\text{Precision} + \textit{Recall}} \text{ where } \beta^2 = \frac{1-\alpha}{\alpha} \tag{4.15}$$

where $\alpha \in [0, 1]$. However, precision, recall and F measure are set-based measures which are computed using unordered sets of documents/sentences. Losada et al. [65] stated that F measure is not very precise in characterizing real requirements of users in the context of sentence retrieval. In our system, the relevant sentences are ranked by their similarity scores. Therefore, we considered other measure to evaluate the retrieval performance: *precision at k* (P@k). P@k is the proportion of retrieved sentences that are relevant to the query within the top k ranked sentences, i.e.:

$$\text{P@k} = \frac{N_r}{k} \tag{4.16}$$

where N_r is the number of relevant sentences in the top k retrieved sentences. In the following experiments, $k = 10$. The P@10 of the corresponding queries in each collection are averaged out to get a single P@10 for each collection.

4.4.4 Evaluation Criteria

The final goal of our SSR system is to provide users with various kinds of sentences relevant to the query. Therefore, in the SR module, we deploy two evaluation criteria: *equivalence criterion* and *relevance criterion*. The former is a strict one, in which only sentences that are similar to the query are regarded as relevant. On the other hand, the latter is a loose one, in which the relevant sentences can also contain other related information to the query. Examples of relevant sentences of the query "*SARS is a worldwide health threat.*" on these two criteria are in Table 4.4. The sentence S_2 is non-relevant on equivalence criterion as it supports the opposite idea of the query (that is SARS is not a health threat) while it is regarded as relevant on the relevant criterion.

Table 4.4: Example sentences of relevant sentences on two criteria

		Equivalence criterion	Relevant criterion
Q:	SARS is a worldwide health threat.		
S_1:	The World Health Organization has issued a worldwide alert on SARS.	YES	YES
S_2:	It has become evident that SARS is no longer a threat on the other side of the river.	NO	YES
S_3:	Other data did reveal, however, that the groundwater could pose a serious threat to human health.	NO	NO

4.5 Experimental Results and Discussion

4.5.1 Results of HySR

Table 4.5 and 4.6 reveal P@10 of our proposed system applying HySR algorithm in comparison with the baseline TF-ISF on equivalence and relevance criterion, respectively. Bold numbers indicate the best performance of each row. Different values of β in the HySR algorithm are examined in order to get a more detail of the influence of syntactic score and the lexical score in the sentence retrieval system. We analyze HySR with β ranges from 0 (only lexical score is considered) to 1 (only syntactic score is considered). Fig. 4.4 and 4.5 illustrate the average P@10 of two methods (TF-ISF and HySR) with different values of β on two criteria. In both criteria, our proposed method (HySR) yields better results than the baseline for all queries. The best performance of HySR is achieved at $\beta = 0.2$. This indicates that lexical matching is more important in retrieving relevant sentences in comparing with syntactic matching. On average, P@10 of our system is 16.91% and 27.45% higher than one of TF-ISF on equivalence criterion and relevance

criterion accordingly. We can conclude that our method considering both syntactic and lexical similarities is effective to retrieve relevant sentences.

Table 4.5: P@10 of HySR and TF-ISF for Equivalence criterion

Collection	Baseline (TF-ISF)	HySR ($\beta = 0$)	HySR ($\beta = 0.2$)	HySR ($\beta = 0.5$)	HySR ($\beta = 0.7$)	HySR ($\beta = 1$)
1990	15.0	11.7	10.0	11.7	**16.7**	15.0
1992	40.0	58.0	**60.0**	**60.0**	54.0	48.0
1993	14.0	**28.0**	**28.0**	26.0	14.0	6.0
1994	24.0	44.0	**46.0**	44.0	38.0	32.0
1998	13.3	28.6	**33.3**	31.7	24.3	16.7
1999	**35.0**	27.5	**35.0**	**35.0**	**35.0**	32.5
2000	15.0	50.0	**51.7**	48.3	48.3	38.3
2001	20.0	**22.5**	20.0	20.0	20.0	17.5
2002	10.0	**28.0**	**28.0**	24.0	20.0	14.0
2003	22.5	42.5	53.8	**55.0**	46.3	36.3
AvgP@10	20.00	34.55	**36.91**	36.00	32.36	25.64
△%		*(+14.55)*	***(+16.91)***	*(+16.00)*	*(+12.36)*	*(+5.64)*

Table 4.6: P@10 of HySR and TF-ISF for Relevance criterion

Collection	Baseline (TF-ISF)	HySR ($\beta = 0$)	HySR ($\beta = 0.2$)	HySR ($\beta = 0.5$)	HySR ($\beta = 0.7$)	HySR ($\beta = 1$)
1990	36.7	36.7	41.7	38.3	**45.0**	38.3
1992	62.0	**96.0**	**96.0**	**96.0**	90.0	90.0
1993	56.0	**72.0**	**72.0**	66.0	52.0	42.0
1994	50.0	76.0	**78.0**	72.0	68.0	62.0
1998	50.0	85.7	**88.3**	85.0	87.1	56.7
1999	50.0	37.5	52.5	52.5	**55.0**	42.5
2000	30.0	66.7	**76.7**	73.3	73.3	63.3
2001	25.0	**27.5**	25.0	25.0	25.0	20.0
2002	16.7	54.0	**60.0**	46.7	40.0	40.0
2003	43.8	68.8	**81.3**	**81.3**	75.0	58.8
AvgP@10	42.00	64.00	**69.45**	66.91	63.26	53.45
△%		*(+22.00)*	***(+27.45)***	*(+24.91)*	*(+21.26)*	*(+11.45)*

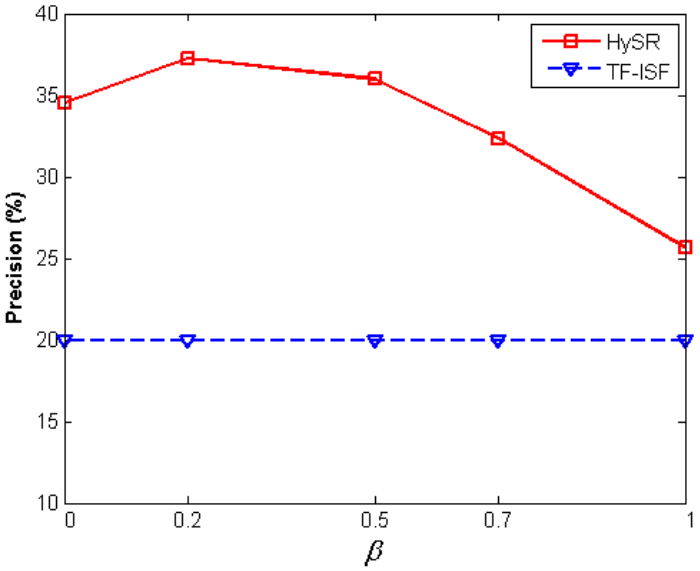

Figure 4.4: Average P@10 on equivalence criterion with different values of β

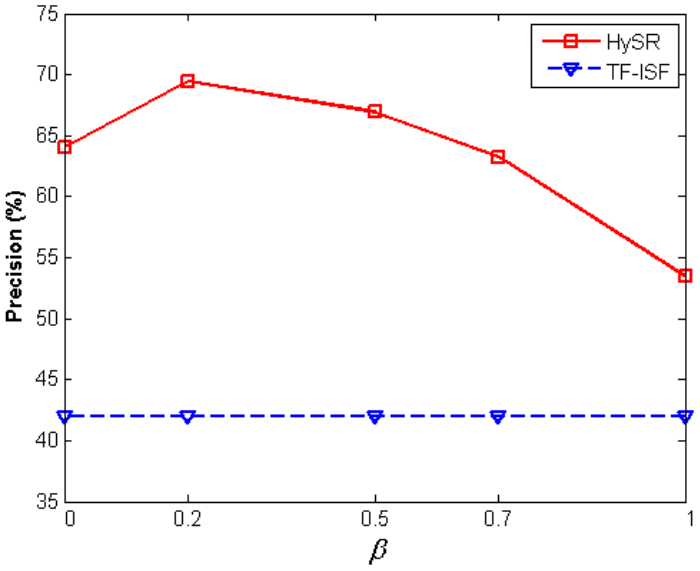

Figure 4.5: Average P@10 on relevance criterion with different values of β

4.5.2 Results of HySR with different weighting schemes

Results of P@10

Table 4.7 and 4.8 show the results of different weighting schemes on HySR system on equivalence and relevance criteria. Bold numbers indicate the best performance of each collection. The results are also plotted into charts as Fig. 4.7 and 4.6. We can see that in most collections, the strongest weighting scheme is the combination of IDF and SPEC. On average, COMB weight improves HySR system with trivial EQUAL weight by 2.36%

for equivalence criterion and 5.27% for relevance criterion. In several cases, COMB gives lower result than EQUAL, but it is still higher than the baseline TF-ISF method (see Fig. 4.6). Results of IDF and SPEC are comparable. SPEC yields higher results in comparison to IDF in most collections in equivalence criterion (7/10 collections) while the opposite phenomenon occurs for the relevance criterion (7/10 collections have higher results when using IDF weight). This may be reasoned by the fact that a popular term usually has general semantic meaning. As a result, SPEC weight has already implied the idea of IDF weight (that is, low IDF term has low specific weight). However, SPEC weight depends on the WordNet resource completely. This leads to cases that WordNet does not have a term t while t appears in set of documents, while IDF can capture the specificity of t easily. This is also the reason that the our proposed combining weighting scheme achieves the best performance for both criteria.

Table 4.7: P@10 of HySR with four different weighting schemes on Equivalence criterion

Collection	EQUAL	IDF	SPEC	COMB
1990	10.0	**16.7**	10.0	15.0
1992	60.0	**62.0**	**62.0**	**62.0**
1993	28.0	24.0	28.0	**30.0**
1994	46.0	44.0	46.0	**48.0**
1998	**33.3**	31.7	**33.3**	**33.3**
1999	35.0	40.0	35.0	**42.5**
2000	**51.7**	**51.7**	**51.7**	**51.7**
2001	20.0	**35.0**	32.5	**35.0**
2002	28.0	26.0	28.0	**32.0**
2003	53.8	51.3	**53.8**	48.8
AvgP@10	36.91	38.00	38.00	**39.27**
△%		*(+1.09)*	*(+1.09)*	***(+2.36)***

Table 4.8: P@10 of HySR with four different weighting schemes on Relevance criterion

Collection	EQUAL	IDF	SPEC	COMB
1990	41.7	53.3	45.0	**55.0**
1992	**96.0**	92.0	**96.0**	**96.0**
1993	72.0	**78.0**	74.0	**78.0**
1994	78.0	**80.0**	78.0	**80.0**
1998	88.3	90.0	88.3	**91.4**
1999	52.5	**75.0**	57.5	**75.0**
2000	**76.7**	**76.7**	**76.7**	**76.7**
2001	25.0	**47.5**	40.0	**47.5**
2002	60.0	58.0	**60.0**	58.0
2003	**81.3**	77.5	**81.3**	78.8
AvgP@10	69.45	73.82	71.45	**74.73**
△%		*(+4.36)*	*(+2.00)*	***(+5.27)***

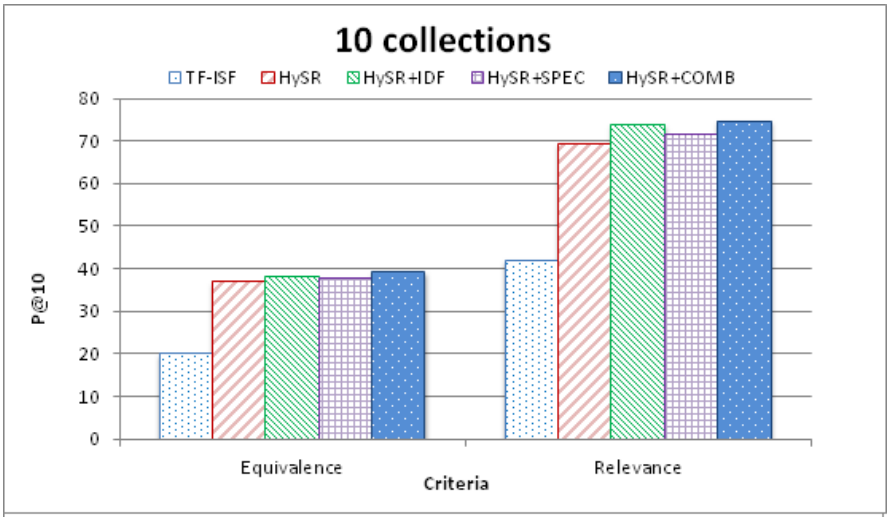

Figure 4.6: Average P@10 on two evaluation criteria

4.6 Summary

In this chapter, we have presented an empirical study of sentence retrieval in the context of support-sentence retrieval. Firstly, the benefit of using the features in a full-sentence query was exploited to enhance the performance of a state-of-the-art sentence retrieval model. We proposed to use a hybrid approach, namely HySR, to retrieve relevant sentences by capturing the similarity of both syntactic and lexical representations of the query sentence and the candidate sentence. Our system which applied HySR algorithm yields a 16.91% higher P@10 compared with the baseline TF-ISF. Performance improves to a 27.45%

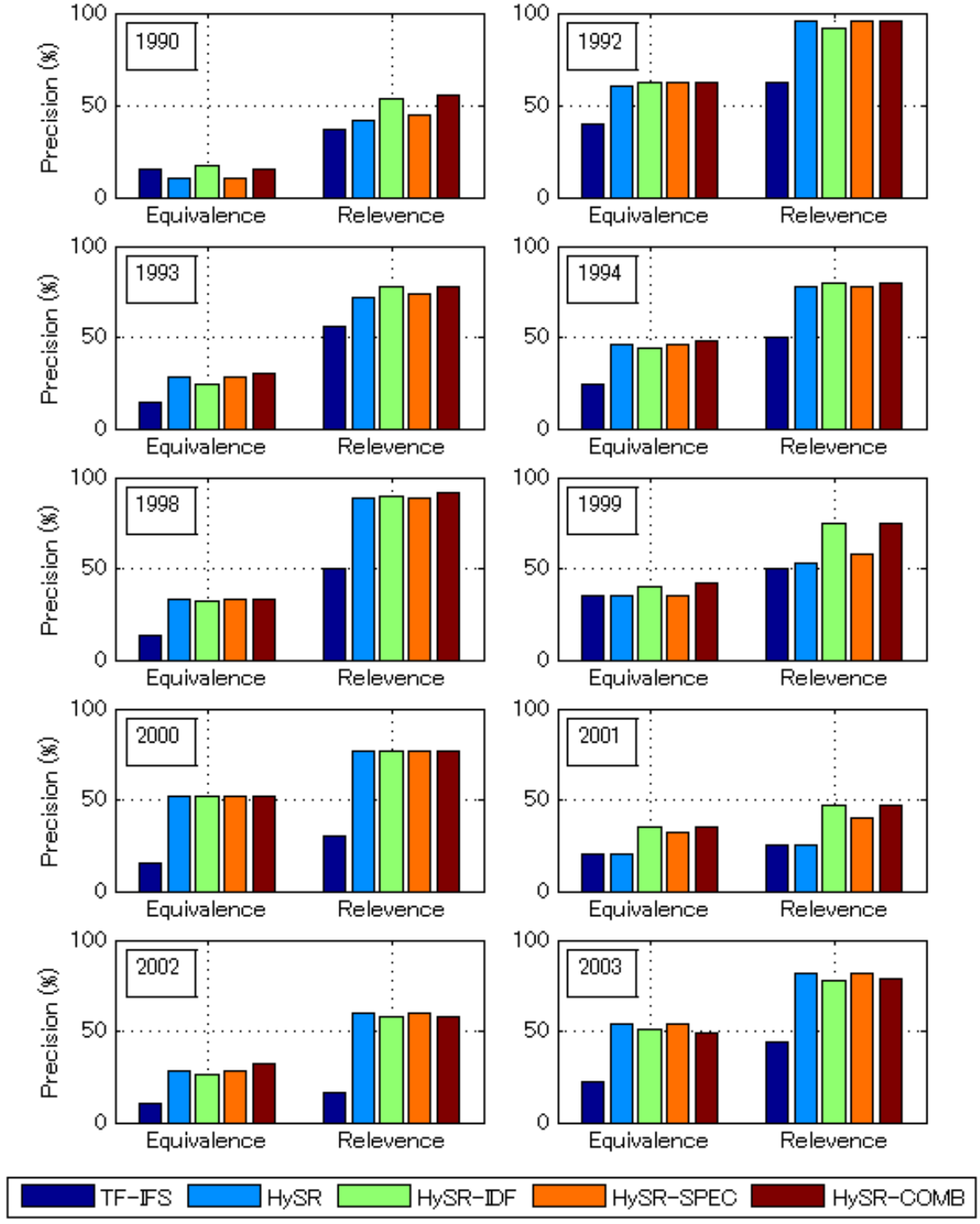

Figure 4.7: P@10 of 10 collections on two evaluation criteria

higher P@10 if we loosen the evaluation criteria.

As a step forward to improve the performance of HySR, we have incorporated HySR

with different query weighting schemes. The results shows that adding weights to query terms can improve the performance of our system. Specifically, our proposed COMB weight is empirically proved to be a robust weighting scheme for recognizing the important terms in the query.

Chapter 5

Impact of Word Sense Disambiguation in Support-Sentence Retrieval

5.1 Overview

The central concept of a sentence retrieval system is the computation of matching between a query and a sentence. However, a sentence usually contains a very limited number of words[1]. This leads to just a few matches that can be found in each pair of query and sentence. An intuitive way to solve this problem is to add up some accordant words to the query so that the possibility of matches will increase. This is called *query expansion*. However, it is not easy to add appropriate terms to the query since doing so can easily introduce noise to the system. In Section 2.2, we have briefly discussed the problem of sense ambiguity in query expansion. One method to address this problem is to disambiguate the query term before expansion. There are some studies on the effectiveness of WSD on document retrieval system. However, to the best of our knowledge, there is no research on this issue for sentence retrieval. Therefore, in this chapter, we study the impact of applying word sense disambiguation into query expansion given the context of support-sentence retrieval. Specifically, we try to answer the question: "Does WSD help enhance the performance of a support-sentence retrieval system?"

In short, the main contribution of this chapter includes:

- Literature review of previous work on applying WSD to IR.
- The first study on the effectiveness of integrating WSD to a SR system.

In the next section, we discuss on previous work on WSD and applying WSD to

[1]Statistics from our corpus reveals that on average, each sentence consists of around 20 words (see Table 2.2).

IR. Section 5.3 presents our proposed system for studying the impact of WSD in SR, which includes the building of a WSD system and the integration of it into SR system. Experimental results and discussions are given in Section 5.5. Finally, Section 5.6 will summarize this chapter.

5.2 Related Work

5.2.1 Word Sense Disambiguation

Words can have more than one distinct meaning. For instance, consider the following sentences[2]:

(1) I can hear *bass* sounds.

(2) They like grilled *bass*.

The occurrences of the word "*bass*" in the two sentences denote different meanings: low-frequency tones and a type of fish, respectively. Among 121 most frequent English nouns (which account for about one in five word occurrences in real text), each word has an average of 7.8 meanings[3]. Although most words are polysemous, it is usually not a problem for a person to understand it clearly (without ambiguity) in real life. However, to a computer, disambiguate a polysemous word automatically is described as "AI-complete" problem [3]. In the field of computational linguistics, the problem is generally called Word Sense Disambiguation (WSD). It is defined as the problem of automatically identifying the meaning (or sense) of a word in a given context [88]. In natural language processing, WSD plays an important role in many applications, such as machine translation, information retrieval, speech processing, etc.

WSD task has been noticed since the late of 1940s [129]. The first experiment by Kaplan proved that just one or two words on both sides of an ambiguous word can be evidence to disambiguate that word [50]. Later, more useful information from context was discovered by numerous work in WSD. Yarowsky introduced simple set of features (context around the ambiguous words) in accent restoration task [133]. This led to many other improved sets of features, such as syntactic dependencies [22, 74, 134], or cross language evidence [34]. So far, there have been total 7 workshops specific on evaluating WSD systems (SENSEVAL-1 in 1998, SENSEVAL-2 in 2001, SENSEVAL-3 in 2004, SEMEVAL-2007, SEMEVAL-2010, SEMEVAL-2012 and SEMEVAL-2013) that contribute greatly to the exploring of effective knowledge sources as well as disambiguation methods. Numerous studies have also been devoted to WSD in languages other than English [54, 91, 94, 115]. Table 5.1 summarizes the main approaches to WSD. According to the knowledge sources used in sense disambiguation, WSD methods are classified as

[2]The examples are taken from the paper of Navigli [88].

[3]Statistics from the Princeton WordNet [82].

Table 5.1: Main approaches to word sense disambiguation

Approach	Technique
Knowledge-based	Mannual disambiguation rules
	Selectional preferences [12, 75, 118]
	Comparing dictionary definitions to the context [19, 61, 125]
	The sense most similar to its context, using semantic similarity measures [35, 78, 99]
	"One-sense-per-discourse" and other heuristics [34, 132]
Unsupervised corpus-based	Cluster word occurrences or contexts, thus inducing senses [56, 63]
	Using an aligned parallel corpus to infer cross-language sense distionctions [16, 17, 33]
Supervised corpus-based	Supervised machine learning, trained on a manually-tagged corpus [60, 89, 133]
	Bootstrapping from seed data (semi-supvevised) [77, 100]
Combinations	Unsupervised clustering techniques combined with knowledge base similarities [103]
	Using knowledge bases to search for examples for training in supervised WSD [80]
	Using an aligned parallel corpus, combined with knowledge-based methods [15]
	Using domain knowledge and subject codes [93]

knowledge-based, unsupervised corpus-based, supervised corpus-based and combinations of these [3]. Comparing and evaluating different WSD systems is extremely difficult due to different test sets, sense inventories and knowledge resources. However, the methods that base on supervised learning are proved to be the best systems. They have been one of the most successful approaches in the last fifteen years in WSD [88]. Therefore, in this study, we would like to apply a supervised learning method to a support-sentence retrieval system and evaluate the effectiveness of WSD in retrieving support-sentences.

5.2.2 Word Sense Disambiguation in Information Retrieval

Current research in IR primarily relies on processing tokens in text, without performing any deeper semantic analysis. Therefore, one possible direction to obtain better performance is to exploit natural language processing to perform semantic analysis of text. Among various semantic annotations, a basic form is word sense annotation. However, does applying WSD to IR help enhance the system's performance? That question is still argued for a complete answer [90]. Most of the early work on the contribution of WSD

to IR resulted in no enhancement at all [25, 111, 121, 126]. There are some reasons for this phenomenon. Firstly, the impact of ambiguity on retrieval effectiveness is small due to the skewed distribution of the senses of many words along with word collection effects [112]. In domains where large numbers of terms in query and documents are common, the benefit of WSD cannot be achieved. In situations where queries are short or terms are used in uncommon senses, ambiguity is a problem and benefits from WSD may be found by answering the question how successful the system is at disambiguating the rare cases where a word is used in an infrequent way. Voorhees [126], Krovetz and Croft [53] concluded that WSD must be very precise on uncommon terms to be able to improve the IR system. However, all of the above researches used simple dictionary (or thesaurus) based word sense representation with relatively small datasets, which resulted in low accuracy of WSD. This is also another reason for the negative results from the early studies on this problem [112].

On the other hand, encourage results have been reported by other studies. The work of Schutze and Pedersen is the first published research that confirmed the improvement of IR performance using an unsupervised word sense disambiguator [113]. Krovetz and Croft also verified the effectiveness of WSD to IR using a manual disambiguation manner [53]. The work of Gonzalo *et al.* [41] and Stokoe *et al.* [119] indicated positive results as well. The main reason of these successes is due to larger training data. Gonzalo even used a manually sense tagged corpus just to test for the effectiveness of WSD in IR. Relaxing the strict disambiguation also helps overcome some negative effects of erroneous disambiguation. Specifically, in certain retrieval tasks, an average accuracy of WSD may yield performance increases [41]. In the meantime, a comprehensive study on the impact of WSD in IR using state-of-the-art WSD models and IR models is still to be done[90]. In the scope of our work, we believe that since sentence retrieval has specific characteristics of information retrieval, an evaluation of the effectiveness of WSD to sentence retrieval is an interesting topic. To the best of our knowledge, there is no literature review on effectiveness of WSD in sentence retrieval. Our research is the first attempt to evaluate WSD in support-sentence retrieval.

5.3 Proposed System

5.3.1 Overview

Both the computations of syntactic score and lexical score in our SR system take into account the synonyms of term q in query Q (Subsection 4.3.1) to ease off the matching between q and terms in a candidate sentence S. This is a kind of query expansion before retrieval. However, since we do not know the correct sense of q, synonyms of all senses of q will be added to the set and may lead to error in the retrieval step. In this chapter,

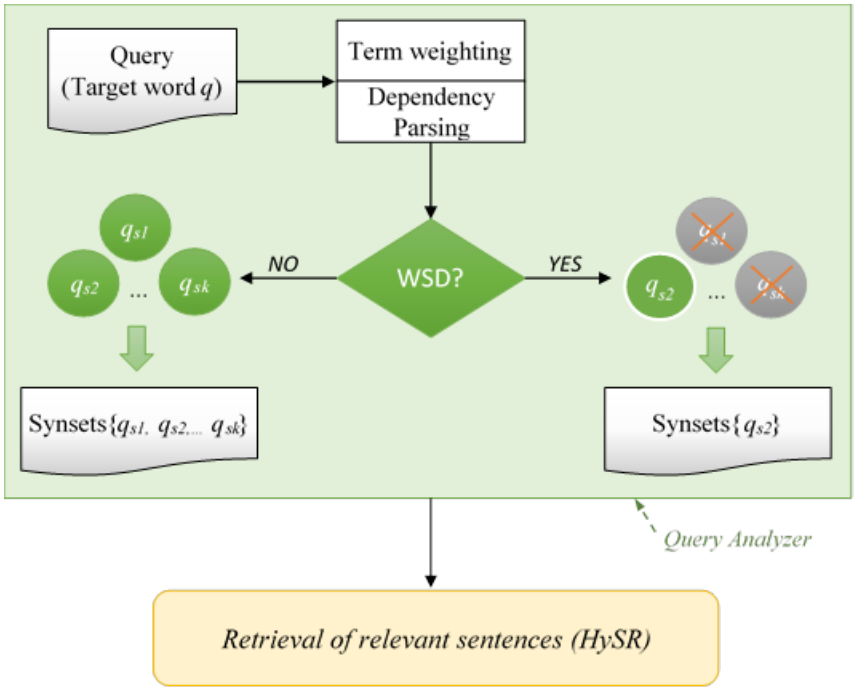

Figure 5.1: SR system work flow with and without WSD in Query Analyzer

we design a system that is able to disambiguate q before query expansion. The system integrates a WSD module into the Query Analyzer process. Fig. 5.1 illustrates the SR system work flow with and without WSD in Query Analyzer. The target word q stands for a polysemous word in Q. q has k meanings (or senses): q_{s_1},q_{s_2},...,q_{s_k}. Suppose that the correct sense of q in Q is q_{s_2}. Applying WSD in Query Analyzer will help eliminate the incorrect senses of q. Therefore, the expanded terms for q contain only synsets of q_{s_2}. The retrieval of relevant sentences is the same with the system using HySR algorithm with combining weighting in Chapter 4. In the next subsections, we will describe our WSD module that applies supervised machine learning and how to integrate this module into SR system.

5.3.2 Support Vector Machines as WSD classifiers

Among supervised methods to WSD, Support-vector machines (SVM) approaches proved to be one of the best systems in several competitions [16, 42, 120]. In this section, we present our system that exploits SVM to learn the WSD classifiers.

Support Vector Machine (SVM) [18] learns a linear discriminant hyperplane that separates two classes of data represented as high-dimensional vectors. SVM is a binary classifier, however, the number of senses for an ambiguous word can be three or more in practice. Therefore, in this research, we follow one-vs-rest multi-class classification approach, in which k SVM models are constructed where k is the number of classes. The i^{th} SVM is trained with all of the examples in the i^{th} class with positive labels, and all

other examples with negative labels.

Let w be the target word (the ambiguous word in a sentence), we encode its surrounding context as a feature vector. The feature set F of w is denoted as in (5.1), where f_i represents a feature.

$$F = \{f_1, f_2, ..., f_n\} \tag{5.1}$$

We consider some textual information as features as follows.

Bag-of-Words

Bag-Of-Words (BOW) feature encodes single words around the target word in a sentence. Therefore, F_{BOW} is a set of all possible words appearing in the context of target instances in the training corpus. Note that only context words in the context are used as BOW features.

POS

This feature encodes part-of-speech of each word in a context window c around the target instance w as in Eq. (5.2), where p_i is the position of the word and P_i is its POS. p_i is an integer in the range $(-c, c)$ indicating the distance between a target word and a word in the context. If p_i is positive, the context word appears in the context after the target word. Similarly, p_i is negative for words in the context before the target word. If p_i exceeds the sentence boundary, P_i is denoted by the null symbol ϵ. For POS feature, F_{POS} is a set of all possible pairs of the position of the word in the context and its POS found in the training corpus.

$$f_i = (p_i, P_i) \tag{5.2}$$

Collocation

Collocation feature (COL) encodes a sequence of words (n-grams) that co-occurs with the target word. Let w_i denote the *i-th* word to the right (or left if i is negative) of the target instance w_0. If the *i-th* word exceeds the sentence boundary, $w_i = \epsilon$. A collocation string is defined as in Eq. (5.3).

$$C_{l,r} = w_l w_{l+1} ... w_r \tag{5.3}$$

For each target instance in the corpus, we extracted 9 collocation strings: $C_{-1,0}$; $C_{0,1}$; $C_{-2,0}$; $C_{-1,1}$; $C_{0,2}$; $C_{-3,0}$;$C_{-2,1}$; $C_{-1,2}$; $C_{0,3}$. Each feature f_i is extracted as in Eq. (5.4), where l_i and r_i are the start and end positions of a collocation string ($1 < r_i - l_i < 4, l_i = -3, ..., 0, r_i = 0, ..., 3$). For the COL feature, F_{COL} is a set of all possible collocation strings with w in the training data.

$$f_i = (l_i, r_i, C_{l_i,r_i}) \tag{5.4}$$

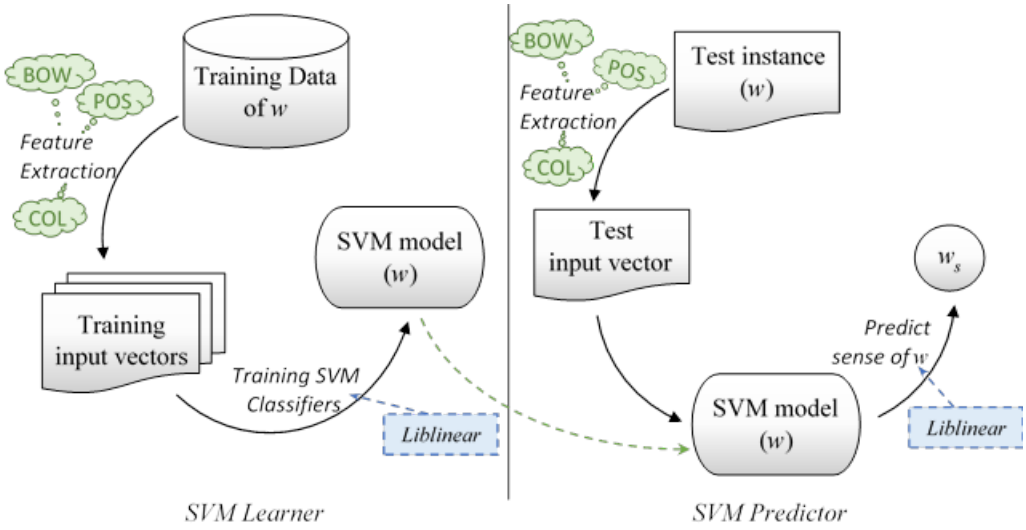

Figure 5.2: The training and test processes of WSD system using SVM

Table 5.2 shows an example sentence and the words that are extracted as features of the ambiguous word "*arm*". The three features of each target word w in a sample sentence are combined together to form an input vector for the SVM to learn (Eq. (5.5)).

$$F = \{F_{\text{BOW}}, F_{\text{POS}}, F_{\text{COL}}\} \tag{5.5}$$

After that, the trained model of w is used to predict the sense of w in the test sentence. These processes are demonstrated in Fig. 5.2. Our system is named as SVM-multi. In this study, we use Liblinear [27] to build the SVM classifiers.

Table 5.2: Example BOW, POS and COL features of the target word *arm* in the sentence: "*The research arm of Wako Securities reports an increasing profits.*"

BOW	$research; arm; wako; securities; reports; increasing; profits$
POS	$(-1, N); (0, N); (2, N); (3, N); (4, V); (6, V); (7, N)$
COL	$(-1, 0, research\_arm); (0, 1, arm\_of)$
	$(-2, 0, the\_research\_arm); (-1, 1, research\_arm\_of); (0, 2, arm\_of\_wako)$
	$(-3, 0, \epsilon\_the\_research\_arm); (-2, 1, the\_research\_arm\_of);$
	$(-1, 2, research\_arm\_of\_wako); (0, 3, arm\_of\_wako\_securities)$

5.3.3 Integrate WSD module into SR system

As we discussed in Subsection 5.3.1, the trained SVM classifier of the ambiguous term q is used to predict its sense in Q. This model is built beforehand by Liblinear using a training dataset introduced in the next section. Then, only the synsets of the predicted sense of w are added into the query rather than adding the synsets of all senses as in Eq.

(4.4). Since we only expand w using the synonyms that are in the correct sense of w, too fine grained disambiguation may limit the number of expanded terms that lead to limited support-sentences will be retrieved. Since we also expect to collect sentences that include broader information than the query itself, using coarse grained sense representation as SVM classes is more appropriate for our system. A coarse grained sense s of q can be expressed by the semantic class of q, in which the senses of q are grouped into sets of similar senses. Table 5.3 reveals these 45 lexicographer names and numbers of all words in WordNet, which we use as semantic classes in this study. For example, the noun "*arm*" has 6 senses in WordNet, but when we map them into semantic classes, the number of coarse grained senses reduces to 3 (see Table 5.4).

5.4 Evaluation Configuration

In this section, we describe two experiments that are conducted to empirical study the impact of WSD to SR:

The first experiment, Experiment A: WSD Evaluation, is presented in Subsection 5.4.1. In this experiment, we construct a WSD system as introduced in Subsection 5.3.2 and validate its effectiveness by using an available training/test set.

The second experiment, Experiment B: WSD in SR Evaluation, is presented in Subsection 5.4.2. The trained classifiers of two target words are used to predict their senses in user's queries in the context of suppor-sentence retrieval.

5.4.1 Experiment A: WSD Evaluation

Dataset

We use SENSEVAL-3 data of English lexical sample task[4] for training and test the effectiveness of our SVM-multi system. The data consists of examples extracted from the British National Corpus (BNC). There are at least two tags per item. Training and test data contain about 60 ambiguous nouns, adjectives and verbs. Table 5.5 shows a summary of the target words in SENSEVAL-3 data. Example extracted from the training data is presented in Fig. 5.3.

Measurement

We evaluate SVM-multi system using Precision (Eq. (5.6)), one of the two measures used in the SENSEVAL-3 tasks. The other one is Recall (Eq. (5.7)). However, since the coverage of our system is 100%, P and R are always equal. Hence, we only report results

[4]The data can be downloaded at http://www.senseval.org/senseval3/data.html

of precision in the next section.

$$P = \frac{\#\text{correct answers provided}}{\#\text{answers provided}} \tag{5.6}$$

Table 5.3: WordNet lexicographer names and numbers

Numbers	Names	Contents
00	adj.all	all adjective clusters
01	adj.pert	relational adjectives (pertainyms)
02	adv.all	all adverbs
03	noun.Tops	unique beginners for nouns
04	noun.act	nouns denoting acts or actions
05	noun.animal	nouns denoting animals
06	noun.artifact	nouns denoting man-made objects
07	noun.attribute	nouns denoting attributes of people and objects
08	noun.body	nouns denoting body parts
09	noun.cognition	nouns denoting cognitive processes and contents
10	noun.communication	nouns denoting communicative processes and contents
11	noun.event	nouns denoting natural events
12	noun.feeling	nouns denoting feelings and emotions
13	noun.food	nouns denoting foods and drinks
14	noun.group	nouns denoting groupings of people or objects
15	noun.location	nouns denoting spatial position
16	noun.motive	nouns denoting goals
17	noun.object	nouns denoting natural objects (not man-made)
18	noun.person	nouns denoting people
19	noun.phenomenon	nouns denoting natural phenomena
20	noun.plant	nouns denoting plants
21	noun.possession	nouns denoting possession and transfer of possession
22	noun.process	nouns denoting natural processes
23	noun.quantity	nouns denoting quantities and units of measure
24	noun.relation	nouns denoting relations between people or things or ideas
25	noun.shape	nouns denoting two and three dimensional shapes
26	noun.state	nouns denoting stable states of affairs
27	noun.substance	nouns denoting substances
28	noun.time	nouns denoting time and temporal relations
29	verb.body	verbs of grooming, dressing and bodily care
30	verb.change	verbs of size, temperature change, intensifying, etc.
31	verb.cognition	verbs of thinking, judging, analyzing, doubting
32	verb.communication	verbs of telling, asking, ordering, singing
33	verb.competition	verbs of fighting, athletic activities
34	verb.consumption	verbs of eating and drinking
35	verb.contact	verbs of touching, hitting, tying, digging
36	verb.creation	verbs of sewing, baking, painting, performing
37	verb.emotion	verbs of feeling
38	verb.motion	verbs of walking, flying, swimming
39	verb.perception	verbs of seeing, hearing, feeling
40	verb.possession	verbs of buying, selling, owning
41	verb.social	verbs of political and social activities and events
42	verb.stative	verbs of being, having, spatial relations
43	verb.weather	verbs of raining, snowing, thawing, thundering
44	adj.ppl	participial adjectives

Table 5.4: Example of fine grained and coarse grained senses of the noun '*arm*'

Fine grained	Coarse grained
arm%1:06:00::	06
arm%1:06:01::	06
arm%1:06:02::	06
arm%1:06:03::	08
arm%1:08:00::	14
arm%1:14:00::	14

Table 5.5: Statistics of target words in SENSEVAL-3 data English lexical sample task

Class	# Words	Avg senses (fine)	Avg senses (coarse)
Nouns	20	5.8	4.35
Verbs	32	6.31	4.59
Adjectives	5	10.2	9.8
Total	57	6.47	4.96

<instance id="activate.v.bnc.00044866" docsrc="BNC">
<answer instance="activate.v.bnc.00044866" senseid="38203"/>
<context>
Quantitative Aspects of Experience There are three fundamental dimensions of quantity in experience : (a) intensity ; (b) (spatial) extensity ; and (c) duration . At a neurophysiological level , the intensity of an experience is typically reflected in the number of neurones <head>activated</head> (the phenomenon of recruitment) and , more specifically , in the firing frequency of the relevant neurones . Extensity (for example the size of a patch of light) usually correlates with the number and spatial distribution of receptors activated . Finally , duration is correlated with the period of time for which the relevant neurones are active .
</context>
</instance>

<instance id="activate.v.bnc.00044869" docsrc="BNC">
<answer instance="activate.v.bnc.00044869" senseid="38201"/>
<context>
Extensity (for example the size of a patch of light) usually correlates with the number and spatial distribution of receptors activated . Finally , duration is correlated with the period of time for which the relevant neurones are active . (I am leaving aside phenomena such as accommodation , whereby a constant stimulus when sustained may <head>activate</head> the nervous system progressively less intensively , with a corresponding reduction in the perceived intensity of the stimulus . These , and other features of adaptation , do not invalidate the underlying conceptual framework .) The earliest psychophysical observations demonstrated a correlation between the intensity of the physical stimulus and subjective reports of the intensity of the resultant experience .
</context>
</instance>

Figure 5.3: Example extracted from the training data of the verb '*active*'

$$R = \frac{\#\text{correct answers provided}}{\#\text{total answers to provide}} \tag{5.7}$$

Comparing Systems

The SENSEVAL-3 English lexical sample task aims to create a framework for the evaluation of corpus-based learning systems that perform Word Sense Disambiguation [79]. In this competition, there are 27 teams participated and submitted a total of 47 systems. In the next section, we will present our WSD experimental results in comparing with 37 supervised systems participating in this task.

For the baseline method, we use the result of the Most Frequent Sense (MFS) heuristic method. The MFS system always returns the sense that has the highest frequencies in the sense inventory (i.e. WordNet).

Evaluation Criteria

The results of Experiment A is reported for three criteria of sense representation: *Fine*, *Coarse 1* and *Coarse 2*.

For Fine and Coarse 1 criteria, we follow exactly the definition of them in SENSEVAL-3, in which coarse grained sense of a word is mapped from a set of fine grained senses. Fig. 5.4 shows a part of sense map that is given by SENSEVAL-3. In this sense map, all fine grained senses which are similar are put in one line (one coarse sense). In coarse sense evaluation, if the output of WSD system is one of the senses that belongs to a group, it is regarded as correct classification.

For Coarse 2 criterion, we follow our definition of coarse grained sense, in which semantic class of a sense is regarded as coarse grained sense (see Subsection 5.3.3).

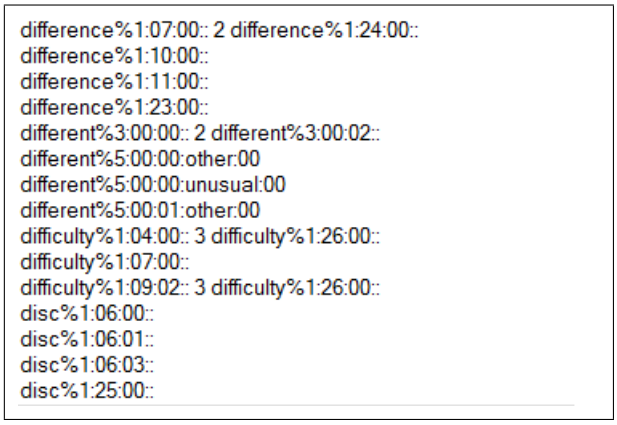

Figure 5.4: Example extracted from the sense map given by SENSEVAL-3

5.4.2 Experiment B: WSD in SR Evaluation

Dataset

In this experiment, we construct a sentence retrieval system that includes WSD module in Query Analyzer process as discussed in Subsection 5.3.1. For training WSD classifiers,

we used the SENSEVAL-3 data from English lexical sample task as discussed above. For evaluation of sentence retrieval, we use the Daily Yomiuri news article corpus as discussed in Subsection 3.1.1. However, in order to evaluate the effectiveness of applying WSD in SR, we need a set of specific queries that satisfy the following criteria:

(1) The query needs to contain a target ambiguous word q.

(2) A WSD classifier for q is available (i.e., SENSEVAL-3 training data must contain q).

(3) Different senses of q occur in the Daily Yomiuri corpus.

These criteria result in a total of 10 queries as given in Table 5.6.

Table 5.6: Queries constructed for Experiment B: WSD in SR evaluation (target words are in bold)

No.	Query	Correct sense ID
1	North Korea is refusal to allow inspections of its suspected nuclear **arms** facilities.	06
2	Russia is subject to a ban on **arms** exports.	06
3	The research **arm** of Wako Securities reports an increasing profits.	14
4	General Noriega had been flirting with the Soviet Union and continues to receive **arms** from Moscow.	06
5	The Government has been resisting the removal of frontier controls because of anxieties about drugs and illicit **arms** traffickers.	06
6	The prefecture **used** to lead the nation in the production and sales of oysters.	41
7	Restaurants serving lamb in Japan **used** to be strictly haute cuisine.	41
8	The concrete and reinforced steel **used** to build the expressway were tested after the earthquake.	34
9	We **used** to think that just one doctor was all we needed to save people's lives.	41
10	Japan **used** to be the largest food donor to North Korea.	41

Evaluation Criteria

In order to investigate an impact of WSD for sentence retrieval, the 10 queries in Table 5.6 are classified from two points of view: (1) C_W or I_W (correctness of WSD) and (2) C_M or I_M (correctness of term matching in sentence retrieval).

Let q be the ambiguous word in the user's query and s be the word in the candidate sentence that is matched to q. If the trained SVM model of q predicts the sense of q correctly, we consider it as correct WSD (C_W), else we denote it as I_W (incorrect WSD). Similarly, C_M and I_M denote the correctness of the matching between q and s in our SR

system. Among the top 10 retrieved sentences, if we could find one or more sentences that contain q and/or its correct synsets, we consider a query as C_M, otherwise we denote it as I_M. The results of each query are therefore reported in two measures: the correctness of WSD classifier in predicting the sense of q and the correctness of SR system in retrieving relevant sentences to Q.

5.5 Experimental Results and Discussion

5.5.1 Experiment A

Table 5.7 shows the precision of our system and two comparing systems: the MFS baseline and the best system in SENSEVAL-3 (htsa3). This result is reported for 3 criteria: *Fine*, *Coarse 1* and *Coarse 2*. Fig. 5.5 and 5.6 present the performance of our system and all 37 systems in SENSEVAL-3 in two representation of senses: *Fine* and *Coarse 1*. These systems also ultilize supervised learning methods. The MFS baseline is also demonstrated in the two figures.

For the fine grained evaluation, our system performs quite good. It ranks 6th in 37 systems and give 17% improvement to the baseline[5]. For the coarse grained evaluation, our system performs acceptable. The system increases 9.6% of precision in comparing to the baseline (*Coarse 1*). However, performance of SVM-multi on *Coarse 2* is much more higher than the one on *Coarse 1*. Specifically, precisions of two SVM-multi classifiers for the two target words in Experiment B are 90% and 100%. Therefore, our SVM-multi system is expected to be appropriate in the investigation of the effectiveness of WSD on SR in Experiment B.

Table 5.7: Precision of our systems in comparison with the MFS baseline and the best system in SENSEVAL-3

Name	Fine	Coarse 1	Coarse 2
MFS	55.2	64.5	70.9
htsa3	72.9	79.3	-
SVM-multi	72.2	74.1	80.8
Δ%	(*+17.0*)	(*+9.6*)	(*+9.9*)

5.5.2 Experiment B

There are 4 cases of results as follows.

[5]Although DLSI-UA-LS-SU system achieves the best precision (78.2% fine grained, 82.8% coarse grained), it is actually the worst system because its recall is the lowest (31.0% fine grained, 32.9% coarse grained).

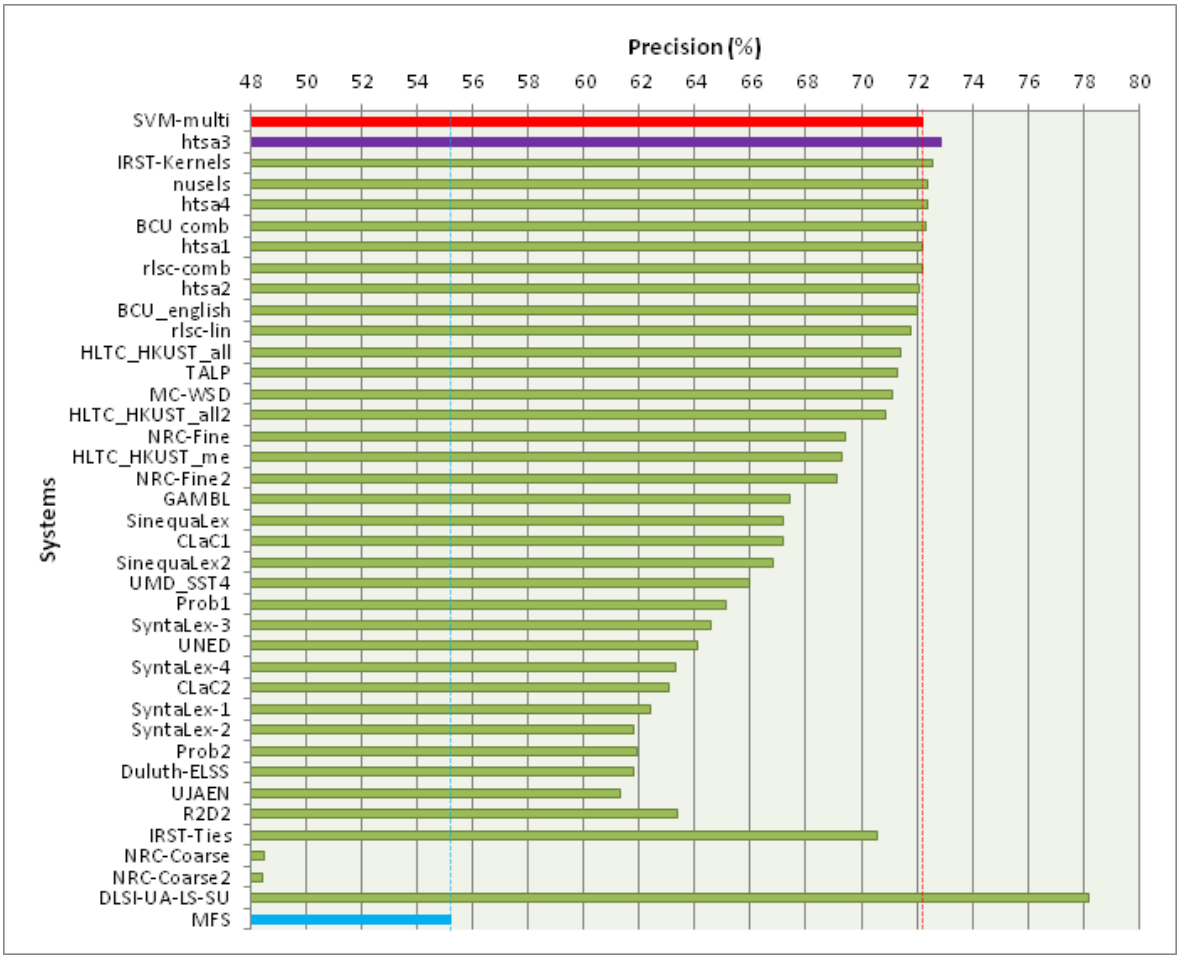

Figure 5.5: Performance of our system in comparison with other systems in SENSEVAL-3 for fine grained scoring

- $C_W C_M$: SVM model predicts the correct sense of q and SR perform the matching correctly. Although we expect this to happen to improve the performance of sentence retrieval, because of the insufficiencies of the context around q, the trained SVM model hardly predicts the correct sense for q. There are only 3 correct WSD classification in 10 queries.

- $I_W C_M$: Although the SVM model predicts the incorrect sense, SR module still correctly match q and s. These are the cases that s and q are matched correctly before query expansion. Therefore, the incorrect senses of q which are added later don't effect the alignment between two sentences. In other words, the context of the candidate sentence and the query sentence have already excluded the possibility of incorrect alignment of the incorrect sense.

- $C_W I_M$: Although the SVM model predicts the correct sense, SR module cannot take advantage of it to retrieve relevant sentences. This happens when s cannot be matched to q before query expansion. When adding some synonyms of w to the

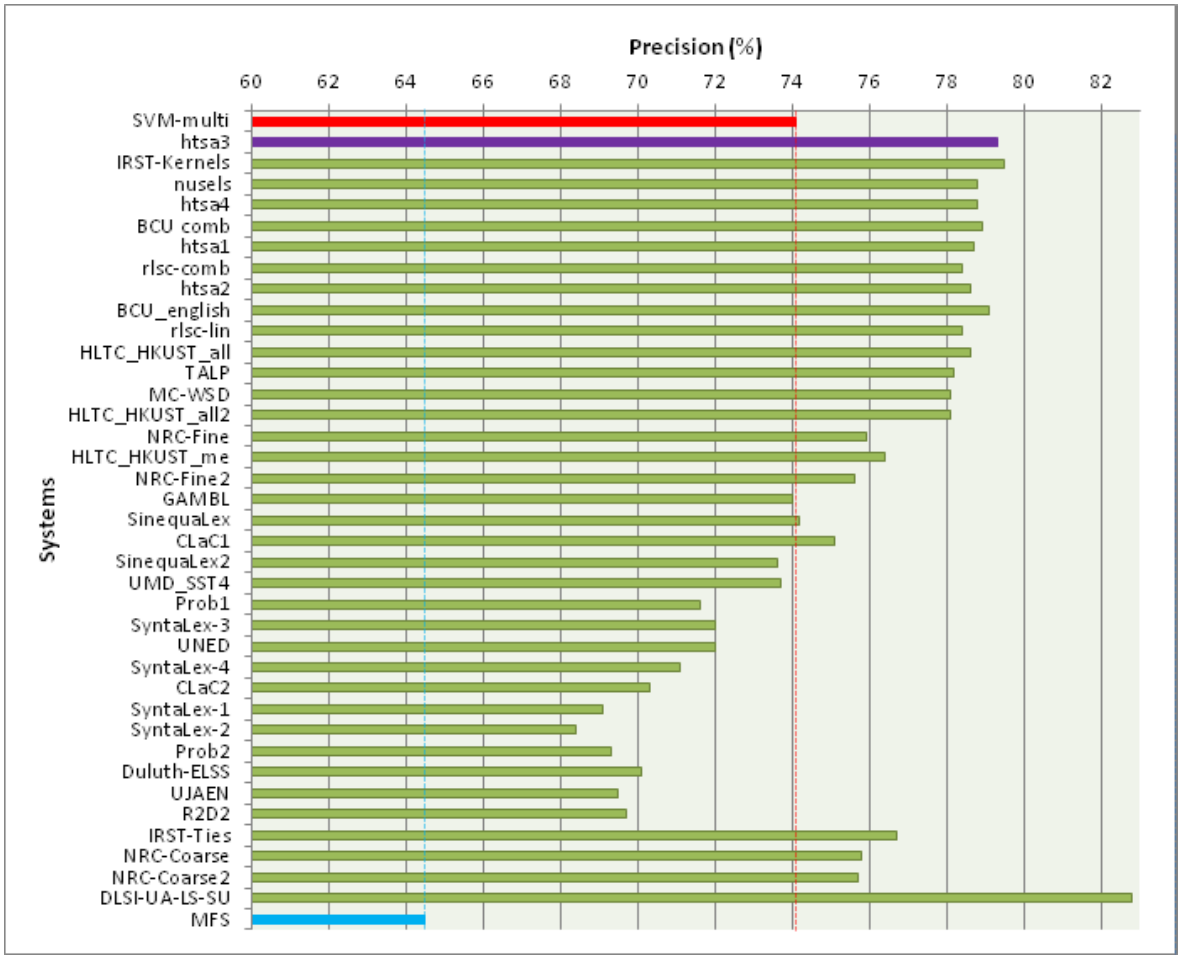

Figure 5.6: Performance of our system in comparison with other systems in SENSEVAL-3 for coarse grained scoring

query, although they are in the correct sense with w, the system still cannot find a word s in the candidate sentence so that s can be matched to one of those terms. This is different with Document Retrieval, in which the context of the candidate document is larger. Therefore, it is easier to find a word s which can be matched to one of the expanded terms of q.

- $I_W I_M$: SR system makes an matching incorrectly because of the wrong ouput of SVM model. In our experiment, we still have not found this case yet. It is maybe because of the number of the test queries is still small.

In summarization, the numbers of each cases above are shown in Table 5.8. We analyze these numbers by computing the percentage of the following two cases:

(1) The percentage of cases when correct WSD and correct SSR co-occur:

$$\frac{C_W C_M}{C_W C_M + C_W I_M} = 67\%$$

(2) The percentage of cases when incorrect WSD and incorrect SSR co-occur:

$$1 - \frac{I_W C_M}{I_W C_M + I_W I_M} = 0\%$$

Table 5.8: Results of WSD in SR evaluation

	C_W	I_W
C_M	2	7
I_M	1	0

These numbers show that applying WSD to SSR seems not so effective. Although the trained SVM model predicts the wrong sense of q, the system still can find the correct relevant sentence. This can be explained by the fact that our proposed method for sentence retrieval uses dependency matching to judge the relevance somehow implies that a kind of disambiguation has been performed for the ambiguous words in the query. For example, the ambiguous word '*arm.n*' is no more ambiguous when it occurs with '*nuclear.arm*'. For those cases, WSD may not help to improve SR performance.

5.6 Summary

In this chapter, we have conducted the SR system which integrated a WSD module to disambiguate an ambiguous word in query. The disambiguation process is performed during the query expansion step. The results show that although WSD helps to select the correct terms to expand to the query, it may not so effect to improve the overall performance of the SR system. The main reason is because of the lack of context information in the sentence as well as in the candidate sentence. It is quite difficult for the SR system to find the matched terms between the query and the candidate sentence although we have removed the noise added by incorrect query expansion. Because of this lack of the context information, the trained SVM model could not perform well, especially in the case that the ambiguous word is used in an infrequent way. However, the number of our test queries are still small to surely confirm the impact of WSD in SR. The results would be more reliable if we extend the test queries as well as evaluate the performance of the SR system with a perfect WSD classifier (i.e. manually tagging).

Chapter 6

Classification of Support-Sentences

This chapter focuses on how to distinguish agreement and contradiction relations between a support-sentence and a query sentence. At first, we introduce the sentence classification problem and some related work. The proposed system and experiments are presented in the followed sections. Discussion is given out at the end of this chapter.

6.1 Overview

Computing has realized human dreams of storing and transferring knowledge, which is primary represented by text. There have been a number of computer systems that can read text as good as a human and search for text much faster than a human does. However, understanding text is still a difficult problem for a computer as natural language is so flexible. Humans can explain an idea in various ways (e.g., using synonym, altering sentence structure) while a computer could not learn such changes as easily as we. In the field of natural language processing, there are many tasks that deal with this issue, such as Word Sense Disambiguation [61], Semantic Role Labeling [39], Classification of Semantic Relations [40], Semantic Textual Similarity [2], etc. Recently, the task to recognize textual entailment (RTE task [21]) between two text fragments has been proposed as a generic task that captures major semantic inference for many NLP applications such as Question Answering, Information Retrieval, Information Extraction and Text Summarization. However, besides entailment, there are more semantic relations among texts. Cross-document Structure Theory (CST) attempts to characterize 18 kinds of relationships that exist between pairs of sentences coming from one or more documents [105]. In our opinion, semantic relations in CST are too difficult to be distinguished automatically. Furthermore, such detail types of semantic relations are not necessary for most of applications, such as our SSR system.

SSR system is designed to support users writing an article on one theme by collecting different ideas (sentences) that are relevant to the topic. The retrieved support-sentences are put into useful semantic categories, which acts as the advantageous hints for their

writing. In this chapter, we consider two most important semantic relations which are agreement and contradiction since they determine the attitude towards the article. If the retrieved sentences are all classified in agreement or contradiction category, the article trends to no controversy. On the other hand, if some retrieved sentences are regarded as agreed to the query but some others are put in contradiction category, there may be various opinions on the topic. Table 6.1 presents some examples on agreement and contradiction sentences of a query.

The main contribution of this chapter is the proposal of five algorithms based on rules and bootstrapping learning technique for the classification module of the SSR system. Most of the previous studies on analyzing the semantic relations between texts applied supervised learning methods to facilitate the semantic features extracted from available annotated corpora [24, 36, 85, 86]. Although the reported results are good, the cost paid for corpus building was too expensive. The bottleneck of the data sparseness is also a serious problem. Our first proposed algorithms, two rule-based methods which require no manually tagged corpus, are demonstrated relatively effective to identify agreement sentences. The contradiction relation is more difficult to identify because it requires more extend knowledge rather than simple word overlaps to recognize the conflicting segments of texts. Our other algorithms based on bootstrapping technique can automatically extract effective clues for identification of contradiction from some initial seed sentences and extend them through iterations. Since the learning process is completely automatic, our approaches require no human intervention.

In the next section, literature review on some previous researches related to our work is presented. The Sentence Classification (SC) framework is described in Section 6.3, in which our proposed algorithms are presented in details. Section 6.4 and 6.5 report the experiment results of our system along with some discussions. Finally, we summarize and point out the future research in Section 6.6.

Table 6.1: Example of agreement and contradiction in SSR system

Query	Rice market in Japan can be opened to foreign countries.
Agreement	Japan has been under strong pressure from other member countries of the General Agreement on Tariffs and Trade (GATT), including the United States, to open its rice market.
Contradiction	Even if the government pledges opening of the nation's rice market to other countries, it might be unable to do so.

6.2 Related Work

One of important related work of sentence classifciation is a new kind of Textual Entailment task called Search task, which was first introduced as a pilot task in RTE-5 [11]. The goal of Search task is to find all sentences in a set of documents that entail a given Hypothesis. The two-way decision between *yes* and *no* entailment in this task is similar to the SR module of our SSR system. However, rather than only extracting entailing sentences, our system retrieves broader information in term of the relevance of sentences. In other words, our system search for sentences that entail different perspectives of the input query (hypothesis). For example, the retrieved sentences can entail the exact query or entail the conflicting idea of the query. Our idea of finding support-sentences is inspired by the Contradiction Detection task of [24, 43, 73, 106, 127], whose studies reported good performance on those training datasets that cover many contradiction phenomena. In contrast, we propose to use unsupervised learning method to acquire contradiction information in text.

Besides RTE challenge, there have been several studies on classifying search results into a fixed set of categories. For example, in sentiment classification and opinion analysis, documents are classified by sentiment polarity (e.g., classification of product reviews into positive and negative categories to evaluate reputation for a product [96, 114, 123]). Our study aims further beyond sentiment classification or opinion analysis because we deal with broader semantic categories rather than positive or negative. The queries in our system are related to not only reputation of products but also facts. For example, the query "*tap water is safe*" is not a matter of positive/negative but a true/false one.

Probably the most closely related work to our study is the research of Murakami *et al.* [85], in which the authors tried to build a classifier to identify semantic relations (agreement, conflict, confinement) between facts and opinions from the internet data. Their system used Support Vector Machine (SVM) classifiers to learn the structural alignments and to detect the semantic relations between sentences on a relatively small corpus (100 documents with around 370 sentence pairs). Mizuno *et al.* [84] developed a similar system to organize information on the web through agreement-conflict relation classification. They used a larger dataset which consists of 1,467 sentence pairs. However, they also exploited a linear classifier which requires training data. Our system, on the other hand, does not require costly hand-tagged training data. In our study, we employed Rule-based Support-Sentence Classifier (RSSC) and Bootstrapping Support-Sentence Classifier (BSSC) to recognize agreement and contradiction semantic classes. We argue that our methods are more potential to be applied in practice.

6.3 Proposed System

6.3.1 Semantic Relations between Sentences

In this subsection, definitions of 5 semantic relations between a query sentence Q and a support-sentence S are introduced. These are the expected outputs of our proposed SSR system. In our point of view, these relations are not separated from each other. In fact, we divide 5 types of relations as in Fig. 6.1, in which we have three main classes: Agreement, Contradiction and Cross-references. The other two classes, Subsumption and Refinement, are subsets of Agreement and Contradiction.

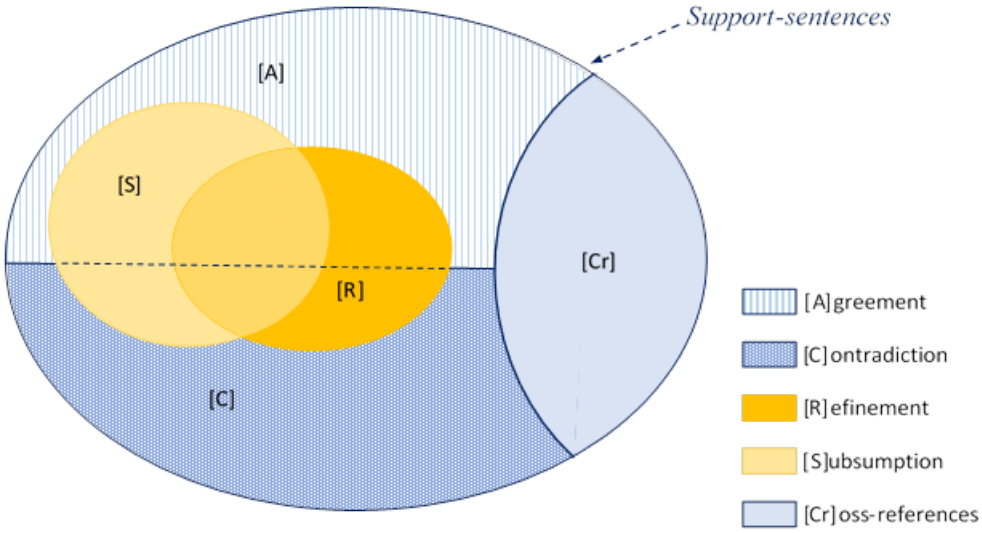

Figure 6.1: Semantic Relations in SSR system

(1) CONTRADICTION

Definition:

S contains or discuss some conflicting information with Q.

Example:

Q:	Japan is positive about Emperor's visiting China for the 20th anniversary of the normalization of Japan-China relations.
S:	Those opposing the trip argue that the Emperor could become involved in diplomatic maneuvering, could be taken advantage of by reformists or conservative forces in China, and could provide the opportunity for China to demand that Japan apologize and pay compensation for its past actions.

In the above example, S discuss on the opposite side of the trip to China of the Emperor, that is not supporting the Emperor to go. Therefore, it is contradict to the idea of the query, which describes the positive attitude towards the visit.

(2) AGREEMENT

Definition:

S contains or discuss the same information with the query and there is no conflicting information with the query.

Example:

Q:	Japan is positive about Emperor's visiting China for the 20th anniversary of the normalization of Japan-China relations.
S:	We have taken the stand of basically agreeing to the idea of the Emperor's trip to China since this year marks the 20th anniversary of the normalization of Japan-China relations.

In the above example, S says that basically they get the agreeing to the idea of the trip to China is *positive*; therefore, S is labelled as Agreement.

(3) CROSS-REFRENCES

Definition:

S mentions the same entity in Q but does not clearly agrees or contrast to Q.

Example:

Q:	Bloods transfusion gives many dangerous viruses.
S:	We offered advice on the design of the necesssary equipment and clean rooms, because this research handles extremely dangerous viruses such as HIV.

In the above example, although S mentions the same entity to Q (dangerous viruses), it is neither agreed nor conflict to the idea in Q. Hence, S is regarded as Cross-References.

(4) REFINEMENT

Definition:

S elaborates or provides details of some information given more generally in Q.

Example:

Q:	Bloods transfusion gives many dangerous viruses
S:	Bloods transfusion gives man hepatitis E.

In the above example, the "*dangerous viruses*" mentioned in Q becomes more specific, "*hepatitis E*" in S. Therefore, the relation between S and Q is Refinement.

(5) SUBSUMPTION

Definition:

S contains all information in Q, plus additional information that is not in Q.

Example:

Q:	Blood transfusion gives many dangerous viruses.
S:	A patient who received a transfusion of blood infected with the HIV virus, which passed through the Japanese Red Cross Society's advanced test in May, has contracted the disease.

In the above example, S provides additional information such as the transfused blood has passed the test of Japanese Red Cross in May. Therefore, S is marked as Subsumption.

In Fig. 6.1, the overlap between Refinement and Subsumption exists, although they are very difficult to distinguish. For example:

Example:

Q:	Blood transfusion gives many dangerous viruses.
S:	Before the advanced test was introduced, three people were reported to have contracted HIV due to transfusions of tainted blood that passed the test used then.

In this example, S refines the "*dangerous viruses*" in Q and also presents additional information to Q (e.g. three people were contracted to HIV, the blood passed the test, etc.)

As we discussed in Section 6.1, in this chapter, only two main classes are considered: agreement and contradiction. Classification of other three classes are remained as future work.

6.3.2 Algorithms for Recognizing Agreement and Contradiction

In this section, we present algorithms to recognize the similarity (Agreement) or dissimilarity (Contradiction) between a pair of sentences, starting with the word overlap, followed by two simple rule-based classifiers and three bootstrapping-based classifiers.

Let Q be a query sentence and S be one of the relevant sentences obtained by the SR module. Note that we suppose S is relevant to Q, i.e., S and Q likely agree or contradict. Both Q and S are represented as a sequence of content words as $Q = (q_1, q_2, ..., q_m)$ and $S = (s_1, s_2, ..., s_n)$. In order to clarify the relevant type between S and Q, they are put through a system similar to the system of Murakami et al. [85], which includes a feature extraction and a semantic relation classification steps. However, unlike Murakami's method, we do not use supervised learning approach to predict the semantic class. All the following algorithms classify candidate sentences into agreement or contradiction, then they are ranked by their scores.

6.3.3 Word Overlap

This method was used as a baseline method in the work of Banea et al. [6]. Despite its simplicity, the method is a relatively effective indicator of sentence similarity and relatedness [71]. Let V_S and V_Q are the binary word vectors that represent S and Q. A weight at each dimension of V_S and V_Q is determined as 1 if the word corresponding to the dimension appears in S or Q, 0 otherwise. In this research, cosine similarity is used as a score between S and Q as in Eq. (6.1). This similarity score is computed for each pair of a candidate sentence and a query sentence, then it is used for ranking the candidate sentences.

$$\text{simscore}(S, Q) = \frac{V_S \cdot V_Q}{|V_S||V_Q|} \tag{6.1}$$

However, this method can only be used for classifying agreement class. Furthermore, two sentences with many overlap words can be totally contradict as follows:

(1) *Blood transfusion may give many dangerous viruses.*

(2) *Blood transfusion gives no danger at all.*

Therefore, we propose to use more reliable approaches for recognizing agreement and contradiction as below.

6.3.4 Rule-based Classifiers

We use two criteria for judging whether a candidate sentence is agreed with or contrast to the query.

(1) Lexical Matching: The lexical matching **lexmatch**(s_i,q_j) evaluates if s_i and q_j have equivalent meaning. In this study, lexmatch(s_i,q_j) = $SS(s_i, q_j)$ defined in 4.3.1: 1 if s_i is a synonym of q_j, otherwise 0.

(2) Negation Clues: We define a binary function **negclue**(s_i, q_j) to indicate if s_i negates q_j or not. This clue can be a polarity mismatch due to the occurrence of negation terms between s_i and q_j, e.g, "*rain*" and "*not rain*"[1]; it can also be an antonym instance, e.g, "*young*" and "*old*". The function value is set to -1 if the negation clue occurs and 1 otherwise.

We use WordNet to identify antonym and synonym of a word. In addition, each term in the query is promoted by a weight value as in Eq. (6.2).

$$\text{fweight}(t) = \begin{cases} 2 & \text{if } t \text{ is a noun} \\ 1 & \text{otherwise} \end{cases} \tag{6.2}$$

We also highlight each pair of words that has the same part-of-speech by an adjustment value δ. δ is set to 2 if s_i and q_j have the same part-of-speech, otherwise $\delta = 1$. Then, each pair of words have a **MATCH** score computed as follows:

$$\text{MATCH}(s_i, q_j) = \delta \times \text{fweight}(q_j) \times \text{lexmatch}(s_i, q_j) \times \text{negclue}(s_i, q_j) \tag{6.3}$$

Algorithm 1 presents all steps of our rule-based classifier RSSC. **Rscore**(S, Q) is defined as sum of MATCH(s_i, q_j) for all pairs of words. If Rscore(S, Q) is positive, S is classified into agreement class, otherwise contradiction. In our study, we evaluate two different RSSC which are denoted as **R** and **R-S**. R is the classifier given by Algorithm 1. While, in R-S, the Rscore(S, Q) is computed as in Eq. (6.4), in which the cosine similarity between two sentences is also taken into account. In this equation, γ is an adjustment parameter and is set to 0.5 in our experiments.

$$\text{Rscore}(S, Q) = \gamma \times \frac{\sum_{i,j} \text{MATCH}(s_i, q_j)}{N} + (1 - \gamma) \times \text{simscore}(S, Q) \tag{6.4}$$

where N is the maximum value of $\sum_{i,j} \text{MATCH}(s_i, q_j)$. Note that both the sum of MATCH and simscore are normalized to [0,1].

[1]More precisely, in this example, both s_i and q_j are "*rain*", but the negation term "*not*" depends on q_j in the sentence Q.

Algorithm 1: RSSC

input : Q: query sentence
U: untagged sentences

output: semantic class for each sentence in U

foreach $S \in U$ **do**
  Rscore $(S,Q) \leftarrow \sum_{i,j}$ MATCH (s_i, q_j)
  if Rscore $(S,Q) < 0$ **then**
    put S to contradiction category
  else if Rscore $(S,Q) > 0$ **then**
    put S to agreement category
  end
end

Sort all sentences in contradiction and agreement categories by their Rscore

6.3.5 Bootstrapping-based Classifiers

Algorithm 2 illustrates our bootstrapping-based classifier BSSC. First, let S_A and S_C be a small amount of seed sentences classified as agreement and contradiction, respectively. The seed sentences of two categories are generated automatically by the following strategies:

(a) *agreement class*: using the query itself as seed or replacing a term in the query by its synonyms;

(b) *contradiction class*: negating the query sentence or replacing a term in the query by its antonyms.

In Algorithm 2, **fsign** evaluates the polarity agreement between two sentences (if two sentences are agreed, fsign is 1, otherwise -1), while **fscore** evaluates the semantic similarity between two sentences but ignoring the polarity. For each unlabeled sentence S, Bscore(S, S_A) and Bscore(S, S_C), the similarity between S and S_A (or S_C), are calculated based on fscore and fsign. If Bscore(S, S_A) is greater than Bscore(S, S_C), S is put into the candidate pool C_{PA}. Then the top n sentences in C_{PA} are added to S_A. Contradiction sentences are also classified in the same way. We repeat the above procedure until no new sentence is classified.

We will present three variations of BSCC: **B-3**, **B-3W** and **B-3SW**. Since the best performance of BSSC is acquired at $n = 3$ (parameter n is the number of newly classified sentences at each iteration) in our preliminary experiments, we always set n to 3 in these bootstrapping-based classifiers. B-3 is the standard BSSC as in Algorithm 2 with $n = 3$. In this basic form, each word is treated equally. However, previous researches have proven that different terms have different important roles in a sentence [67]. Therefore, we propose the second classifier B-3W that uses combining weighting to recognize the important terms in a sentence as described in Subsection 4.3.2. Therefore, the fweight(t)

in Eq. (6.2) is revised as in Eq. (6.5).

$$\text{fweight}(t) = \begin{cases} 2 \times w(t) & \text{if } t \text{ is a noun} \\ w(t) & \text{otherwise} \end{cases} \tag{6.5}$$

where $w(t)$ is the weight of the term t which is computed as in Eq. (6.6) using combining weighting of Eq. (6.7).

$$w(t) = \begin{cases} 0 & \text{if } t \text{ is a stop word} \\ f(t) & \text{otherwise} \end{cases} \tag{6.6}$$

$$f(t) = \text{COMB}(t) = \begin{cases} \text{SPEC}(t) & \text{if } t \text{ is in WordNet} \\ \text{IDF}(t) & \text{otherwise} \end{cases} \tag{6.7}$$

$\text{COMB}(t)$ is the weighting method for sentence retrieval task that we have already proposed in 4.3.2. That is, a weight of a term is defined in terms of the specificity of the term ($\text{SPEC}(t)$) or inverse document frequency ($\text{IDF}(t)$).

In the third classifier B-3SW, to reduce the errors of the lexical alignment step, we also append the cosine similarity score to the fscore in Algorithm 2 as follows:

$$\text{fscore}(S, Q) = \gamma \times \frac{\sum_{i,j} |\text{MATCH}(s_i, q_j)|}{N} + (1 - \gamma) \times \text{simscore}(S, Q) \tag{6.8}$$

where γ is an adjustment parameter and is set to 0.5 in the experiments; N is the maximum value of $\sum_{i,j} |\text{MATCH}(s_i, q_j)|$.

Algorithm 2: BSSC

input : U: untagged sentences

S_A, S_C: seed sentences of agreement and contradiction categories

C_{PA}, C_{PC}: candidate pools of agreement and contradiction categories

output: semantic class for each sentence in U

$i \leftarrow 1$

repeat

  foreach $S \in U$ **do**

    Bscore $(S,S_A) \leftarrow 0$

    Bscore $(S,S_C) \leftarrow 0$

    foreach $A \in S_A$ **do**

      fscore $(S,A) \leftarrow \sum_{i,j} |\ \mathsf{MATCH}\ (s_i, w_{aj})|$

      fsign $(S,A) \leftarrow \prod_{i,j}$ sgn $(\mathsf{MATCH}\ (s_i, w_{aj}))$[2]

    end

    foreach $C \in S_C$ **do**

      fscore $(S,C) \leftarrow \sum_{i,j} |\ \mathsf{MATCH}\ (s_i, w_{cj})|$

      fsign $(S,C) \leftarrow \prod_{i,j}$ sgn $(\mathsf{MATCH}\ (s_i, w_{cj}))$

    end

    Bscore $(S,S_A) \leftarrow \max_{A \in S_A}\{\mathsf{fscore}(S,A) | \mathsf{fsign}(S,A) > 0\}$

    Bscore $(S,S_C) \leftarrow \max_{C \in S_C}\{\mathsf{fscore}(S,C) | \mathsf{fsign}(S,C) > 0\}$

    if Bscore *(S,S_C)* > Bscore *(S,S_A)* **then**

      Add S to C_{PC}

    else if Bscore *(S,S_A)* ≥ Bscore *(S,S_C)* **then**

      Add S to C_{PA}

    end

  end

  Sort sentences in C_{PA} and C_{PC} by Bscore

  Assign agreement and contradiction tag to top n sentences in C_{PA} and C_{PC} and put them to S_A, S_C

  if *there is no new sentences added to* S_A, S_C **then**

    stop $\leftarrow$ true

  end

  $i \leftarrow i + 1$

until stop

[2] sgn(x) is a sign function that returns 1 (if x is positive) or -1 (if x is negative).

6.4 Evaluation Configuration

6.4.1 Dataset

In the experiments of this chapter, we also use the document collection described in Subsection 3.1.1, which consists of 10 collections of Daily Yomiuri newspaper from 1990 to 2003. We prepared 55 queries for each collection and retrieved relevant sentences for those queries by SR module in Section 4.3. These relevant sentences are then be classified into agreement and contradiction categories by methods in Subsection 6.3.2. The number of queries used for evaluation is shown in Table 6.2. The second column 'Agreement' and third column 'Contradiction' indicates that number of queries where some sentences are classified as agreement or contradiction class by our proposed methods. In these experiments, we can obtain agreement sentences for all 55 queries. While contradiction sentences are obtained for only 19 queries. No contradiction sentence is extracted for the rest of 36 queries. The results of different classification algorithms for these queries are evaluated and compared for each collection as well as the whole document collection.

Table 6.2: Number of queries used in sentence classification experiments

Collection	Agreement	Contradiction
1990	6	2
1992	5	3
1993	5	3
1994	5	2
1998	7	3
1999	4	1
2000	6	4
2001	4	0
2002	5	0
2003	8	1
Total	55	19

6.4.2 Measurement

In our SC module, the support-sentences for each classes are ranked by their scores. Therefore, in the experiments, we also use *precision at k* (P@k) as in the SR module. P@k is the proportion of support-sentences that are correctly classified within the top k ranked sentences, i.e.:

$$P@k = \frac{N_c}{k} \tag{6.9}$$

where N_c is the number of correct classified sentences in top k support-sentences in each category. In the following experiments, we analyze the results from $k = 1$ to $k = 20$.

6.5 Experimental Results

6.5.1 Results of 10 collections

Table 6.3 and 6.4 reveals P@10 of recognizing agreement and contradiction categories for 10 collections using different classification algorithms discussed in Subsection 6.3.2. The rows of 2001 and 2002 are omitted in Table 6.4, since no contradiction sentence is obtained for the queries in these two years. Bold numbers indicate the best performance of each collection. In Table 6.3, $\Delta\%$ shows improvement against Word Overlap baseline. Since the baseline is not applicable for categorization of contradiction, $\Delta\%$ in Table 6.4 shows improvement against our rule-based method R. All of our proposed algorithms (RSSC and BSSC) yield better results against the baseline in all queries for agreement class. On average, our best system's P@10 is 2.91% higher than the Word Overlap baseline. This system (B-3SW) also gives the best performance on contradiction class.

Fig. 6.2 and 6.3 demonstrate the average P@10 of the described algorithms in agreement and contradiction class, respectively. In overall, the results in agreement class is much higher than the contradiction class. As shown in Table 6.2, among 55 queries, both agreement and contradiction sentences are extracted only for 19 queries. We can regard that these 19 queries are associated with the controversy topics; there are both positive

Table 6.3: P@10 of Agreement class

		RSSC		BSSC		
Collection	Word Overlap	R	R-S	B-3	B-3W	B-3SW
1990	18.3	20.0	**23.3**	20.0	18.3	21.7
1992	66.0	74.0	72.0	64.0	68.0	**82.0**
1993	30.0	30.0	**32.0**	28.0	28.0	**32.0**
1994	50.0	**52.0**	48.0	34.0	48.0	46.0
1998	28.6	24.3	30.0	22.9	27.1	**40.0**
1999	20.0	**27.5**	**27.5**	25.0	**27.5**	25.0
2000	53.3	43.3	53.3	43.3	**56.7**	46.7
2001	35.0	27.5	35.0	25.0	35.0	**47.5**
2002	20.0	22.0	24.0	12.0	20.0	**26.0**
2003	62.5	58.8	**66.3**	60.0	45.0	53.8
AvgP@10	39.64	38.73	42.36	34.73	37.64	**42.55**
$\Delta\%$		*(−0.91)*	*(+2.73)*	*(−4.91)*	*(−2.00)*	*(+2.91)*

Table 6.4: P@10 of Contradiction class

	RSSC		BSSC		
Collection	R	R-S	B-3	B-3W	B-3SW
1990	0.00	0.00	5.00	5.00	**10.00**
1992	3.33	3.33	10.00	**13.33**	3.33
1993	0.00	0.00	13.33	20.00	**36.67**
1994	**5.00**	**5.00**	0.00	**5.00**	0.00
1998	0.00	0.00	0.00	**10.00**	**10.00**
1999	0.00	0.00	0.00	**10.00**	0.00
2000	12.50	12.50	7.50	10.00	**15.00**
2003	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**
AvgP@10	4.74	4.74	7.89	13.16	**16.84**
Δ%			(+*3.16*)	(+*8.42*)	(+*12.11*)

and negative opinions about these queries. Seeing extracted sentences, we found most of the conflicts is due to the mismatch information caused by news updating. Therefore, the number of contradiction sentences might be much fewer than that of agreement sentences in the document collection. On the other hand, our error analysis discloses that the precision of contradiction class is low because the contradiction between two sentences occurs without recognizable negation clues, which is beyond the reach of our system. This error results in the incorrect computation of **fsign** between two sentences. Table 6.5 gives some error examples of our SC module for the query "*Rice market in Japan is opened to foreign countries.*" The sentences in this table are classified as agreement by B-3SW algorithm but they are actually contrast to the query. For example, in sentence 2, the system could not map the negation clue 'not open' (which is generated automatically in the seed initialization step) with the correct tokens 'unable to do so'. In order to recognize such kinds of contradiction clues, we need more sophisticated system that is able to integrate more diverse knowledge.

6.5.2 Comparison among the algorithms

Table 6.6 shows number of queries for which each system achieved the best performance. When two or more algorithms yield the same results, all the best ones are counted. These numbers support our discussion above, in which B-3SW is the best system. From another point of view, although the agreement results of B-3 and B-3W in Table 6.3 are lower than the Word Overlap baseline, the numbers of best performance cases given by B-3 and B-3W are nearly equal to that of the baseline. They are even higher than R algorithm although R yields better results than B-3 and B-3W in Table 6.3. For the contradiction class, the bootstrapping-based classifiers completely overcome the rule-based classifiers

Table 6.5: Examples of errors in contradiction class for the query: "*Rice market in Japan is opened to foreign countries.*"

No.	Sentence
1	Although some politicians now admit the need for Japan to open its rice market, political parties are reluctant to do this and continue to insist on self-sufficiency in rice because of the House of Councillors election looming next July.
2	Even if the government pledges opening of the nation's rice market to other countries, it might be unable to do so.
3	The decision was aimed at demonstrating Japan's opposition to the opening of its rice market.

in all tables. This verifies the effectiveness of our bootstrapping-based algorithms in recognizing contradiction class.

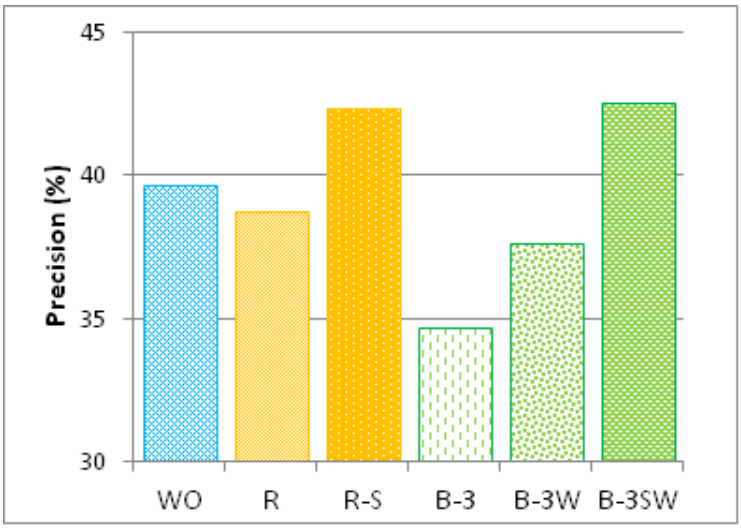

Figure 6.2: Average P@10 of Agreement class

Table 6.6: Best perfomance analysis

	Agreement	Contradiction
Word Overlap	15	-
R	12	3
R-S	19	3
B-3	15	4
B-3W	13	**7**
B-3SW	**22**	**7**

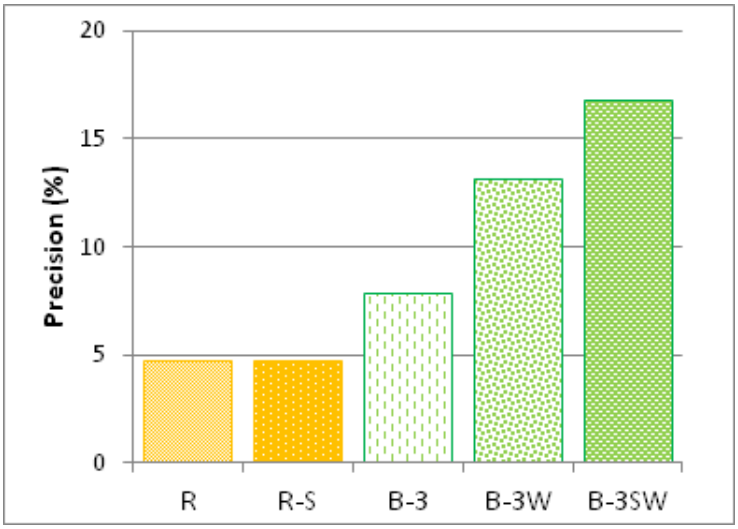

Figure 6.3: Average P@10 of Contradiction class

6.5.3 Analysis of different P@k

In this subsection, sevearal methods are compared according to P@k of different k. For the agreement class, we choose the best classifiers of RSSC and BSSC, which are R-S and B-3SW. The results are plotted in Fig. 6.4. In this figure, we can see that the results of bootstrapping-based algorithm decrease faster than those of rule-based algorithm when k increases. This is because the presence of new seed sentences after each iteration in the bootstrapping-based algorithm have introduced more noise into the agreement category, which leads to the incorrect extraction of new agreement sentences during the learning process. Nevertheless, both R-S and B-3SW perform better than the baseline in all variations of k. For the contradiction class, all three types of BSSC are compared to the best classifier of RSSC as in Fig. 6.5. It is clearly shown that the enhancement of B-3W is achieved by elevating the important terms in a sentence. The improvement is much more higher when we take the cosine similarity between two sentences into account. Once again, the bootstrapping-based classifiers outperformed the rule-based classifier.

6.6 Summary

In this chapter, we have proposed two classifiers (RSSC and BSSC) for the recognition of two semantic relations between sentences in an SSR system, that are agreement and contradiction. Different configurations of these two classifiers are conducted and analyzed using our self-constructing corpora. The B-3SW is confirmed by experiment to yield the best performance on extracting agreement and contradiction sentences. The best classifier is 2.9% higher than the baseline on the agreement class, and 12.1% higher than the best rule-based classifier on the contradiction class. Given the fact that our bootstrapping-based classifiers are trained from untagged data, we believe that applying bootstrapping

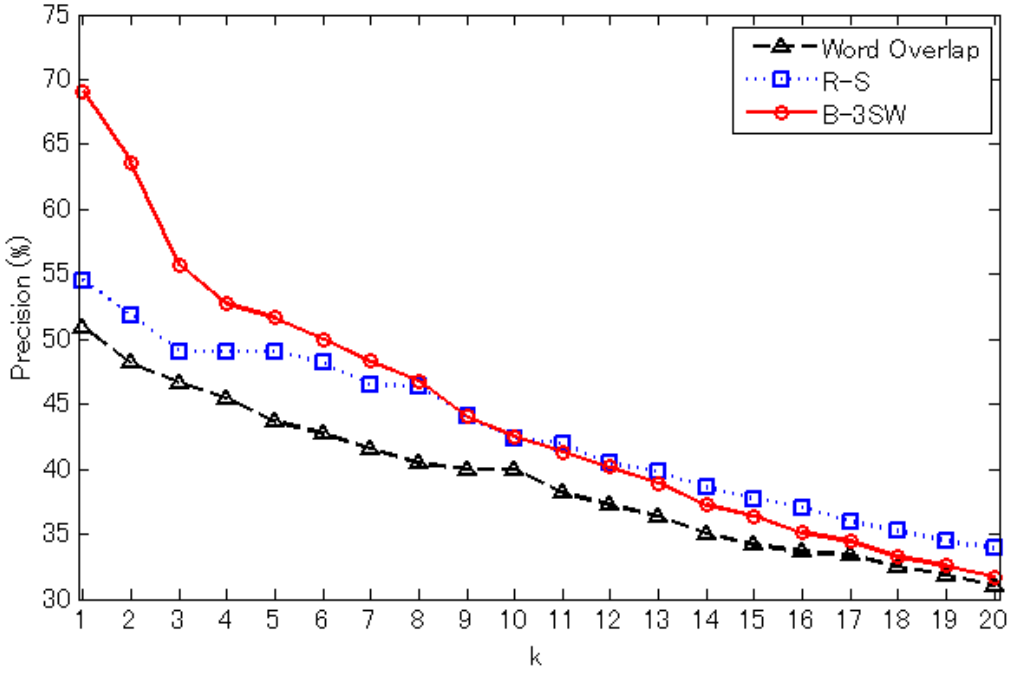

Figure 6.4: P@k with different k in Agreement category

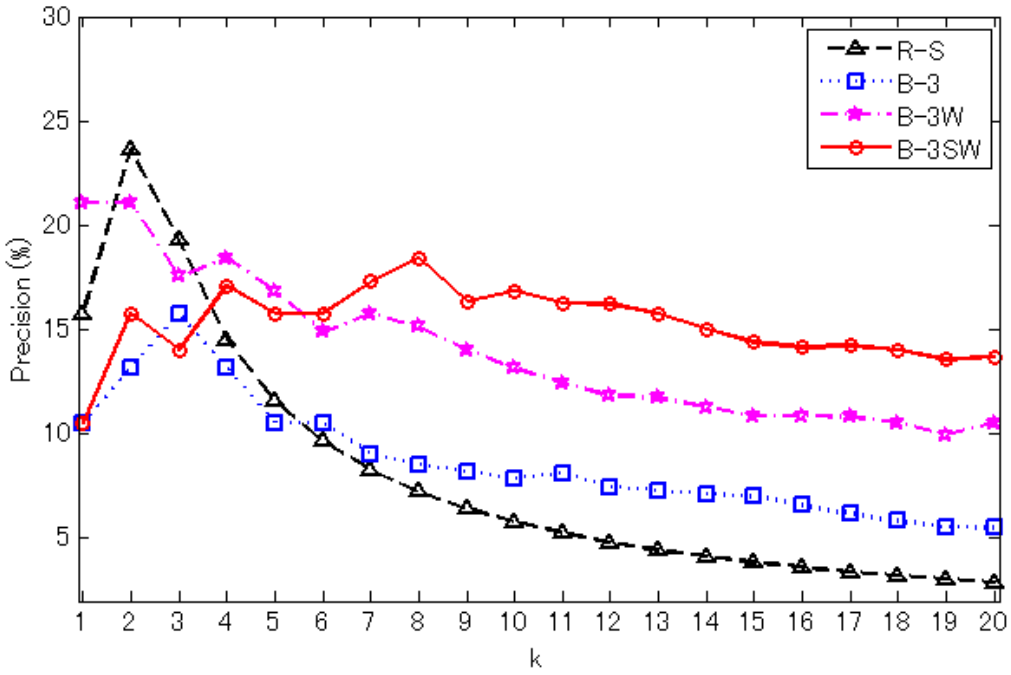

Figure 6.5: P@k with different k in Contradiction category

learning into the classification module of SSR system is promising. Nevertheless, the extraction of contradiction sentences is still a problem in our algorithms due to the complexity of different types of controversy. In the future, we would like to explore more

effective ways to learn the characteristic of contradiction sentences in order to acquire better performance on the contradiction class. Moreover, we would also move forward to the next step of our SSR system, that is recognizing more specific semantic classes such as refinement, subsumption and cross-references.

Chapter 7

Conclusion

This chapter concludes our dissertation by summing up what have been done and what contributions were achieved. At the end of this chapter, we discuss some limits of the study and point out suggestions for future research.

7.1 Summary of the thesis

In this dissertation, we provided a detail investigation of unique characteristics of sentence retrieval. Specifically, we have presented a study on the task of retrieving support-sentences. This study aims to develop a support-sentence retrieval (SSR) system that retrieves sentences relevant to a topic (query), then classifies those sentences into two semantic categories: agreement and contradiction. Thereby, the system helps users quickly navigate through different aspects of the topic. Throughout the thesis, we investigated SSR system step by step. The whole framework of the system consists of three modules:

(1) *Document Retrieval (DR) module*

In this module, we constructed a document retrieval system that employs a vector space model with TF-IDF weighting. Given a set of 10 document collections, we retrieved relevant documents for 55 queries by this system. The output of this module is a ranked list of relevant documents for each query (Chapter 3).

(2) *Sentence Retrieval (SR) module*

Relevant sentences are assumed to be appeared in relevant documents. Therefore, SR system searches for sentences that are relevant to the query among the set of candidate sentences coming from relevant documents. The output of this module is a set of support-sentences. Five different sentence retrieval systems that utilize TF-ISF, HySR and HySR with different query term weightings have been constructed to retrieve support-sentences (Chapter 4). We have also conducted experiments to

investigate the effectiveness of applying Word Sense Disambiguation to the best SR system (HySR+COMB) (Chapter 5).

(3) *Sentence Classification (SC) module*

In this module, the support-sentences from SR system are classified into groups of similar semantic relations. We consider two types of relations: agreement and contradict. We have presented six algorithms including Word Overlap, R, R-S, B-3, B-3W and B-3SW to recognize these semantic relations (Chapter 6).

7.2 Contributions

The main contribution of this dissertation is the introduction to a new task called Support-Sentence Retrieval (SSR). We have designed an overall framework for SSR system and investigated for effective methods to develop two main modules of the system: SR and SC. The main contributions are achieved through the study of these modules.

- For the first task (*SR task*), we proposed a hybrid approach (**HySR**) that utilized both lexical and syntactic information of a sentence to capture the similarities between a query and a candidate sentence. In our study, we exploit the benefit of using a full-sentence as query. That is, besides the lexical matching, grammatical relationships between query terms are taken into account in the process of matching query and candidate sentences. We showed that our proposed method is effective in retrieving support-sentences in comparing with the state-of-the-art TF-ISF method.

- To enhance performance of HySR, we proposed new weighting schemes to capture the importances of different terms in the query (**SPEC** and **COMB**). The experiments of the system that applied HySR along with these weighting schemes yielded additional improvement to the performance of SR system. The best system to retrieve relevant sentences is **HySR+COMB**.

- The next contribution of SR task is a study on the impact of Word Sense Disambiguation (WSD) in sentence retrieval. Literature on sentence retrieval did not provide an answer to solve the sense ambiguity problem in query expansion. Our study is the first attempt to integrate a WSD classifier based on SVM learning to SR system. We showed that at the moment, applying SVM as WSD classifier to SR is not as effective as expected. It is because the lack of information in context of a sentence leads to incorrect classification. In the correct WSD cases, the improvement of SR system is still very limited.

- For the second task (*SC task*), we have proposed two classifiers (RSSC and BSSC) for the recognition of two semantic relations, agreement and contradiction, between

sentences in an SSR system. Different configurations of these two classifiers are conducted and analyzed using our self-constructing corpora. The B-3SW is confirmed by experiment to yield the best performance on extracting agreement and contradiction sentences. Given the fact that our bootstrapping-based classifiers are trained from untagged data, we believe that applying bootstrapping learning into the classification module of SSR system is promising.

7.3 Future Research

The research in this dissertation can be extended in many directions. First of all, in the sentence retrieval module, we applied dependency matching between sentences. However, in natural language, there are numerous alternative ways to express the same idea. These alternative sentences can easily break the dependencies between terms in the original sentence. For example, a passive-active alternation makes the retrieval of relevant sentences difficult. One can be interested in other ways to compute syntactic similarity. For instance, using non-strict dependency matching (e.g. allow partial matching [95]).

On another aspect, although our proposed weighting scheme that utilizes WordNet achieved good performance, it highly depends on the available of extended resources. There are demands for more researches to find out other effective ways to capture the important terms without the requirement of knowledge resources. For instance, discovering important terms based on its dependencies.

For the study on the impact of WSD in SR, currently our study reported negative results. The main reason is because of the lack of information in the sentence that made it difficult for the SVM classifier to predict the sense correctly. Alternatively, one can use manually tagged senses to investigate for the effectiveness of query term disambiguation in a perfect condition. In addition, more large scale empirical studies are needed to surely confirm the performance improvement of a SR system with and without WSD.

For the final task to classify support-sentences into semantic categories, currently our proposed bootstrapping algorithms achieved good precision. However, the running time is quite slow due to the high computational cost in each iteration. A future investigation for improving the algorithm running time is necessary.

The recognition of contradiction sentences in our system is still a challenge due to the complexity of different types of controversy. A future direction is to investigate the characteristic of different contradiction phenomena in order to acquire better performance on the contradiction class.

The number of semantic classes in the current SSR system is two. However, we have presented five types of semantic relations in order to provide users a comprehensive view of a given topic. One important direction of our future plan is to explore effective ways to annotate all these five semantic relations automatically.

Finally, a comprehensive evaluation cannot be done without a gold standard corpus. In future, we plan to construct a manual tagged corpus to give more trustfulness to our results. Once we have been successful in developing a comprehensive SSR system, we could think to extend our system to multi-languages SSR system, which can support not only English but Vietnamese, Japanese, etc.

Appendix A

List of Stopwords

a	anything	but	empty	further
about	anyway	by	enough	get
above	anywhere	call	etc	give
across	are	can	even	go
after	around	cannot	ever	had
afterwards	as	cant	every	has
again	at	co	everyone	hasnt
against	back	computer	everything	have
all	be	con	everywhere	he
almost	became	could	except	hence
alone	because	couldnt	few	her
along	become	cry	fifteen	here
already	becomes	de	fify	hereafter
also	becoming	describe	fill	hereby
although	been	detail	find	herein
always	before	do	fire	hereupon
am	beforehand	done	first	hers
among	behind	down	five	herself
amongst	being	due	for	him
amoungst	below	during	former	himself
amount	beside	each	formerly	his
an	besides	eg	forty	how
and	between	eight	found	however
another	beyond	either	four	hundred
any	bill	eleven	from	i
anyhow	both	else	front	ie
anyone	bottom	elsewhere	full	if

in	nevertheless	same	there	we
inc	next	see	thereafter	well
indeed	nine	seem	thereby	were
interest	no	seemed	therefore	what
into	nobody	seeming	therein	whatever
is	none	seems	thereupon	when
it	noone	serious	these	whence
its	nor	several	they	whenever
itself	not	she	thick	where
keep	nothing	should	thin	whereafter
last	now	show	third	whereas
latter	nowhere	side	this	whereby
latterly	of	since	those	wherein
least	off	sincere	though	whereupon
less	often	six	three	wherever
ltd	on	sixty	through	whether
made	once	so	throughout	which
many	one	some	thru	while
may	only	somehow	thus	whither
me	onto	someone	to	who
meanwhile	or	something	together	whoever
might	other	sometime	too	whole
mill	others	sometimes	top	whom
mine	otherwise	somewhere	toward	whose
more	our	still	towards	why
moreover	ours	such	twelve	will
most	ourselves	system	twenty	with
mostly	out	take	two	within
move	over	ten	un	without
much	own	than	under	would
must	part	that	until	yet
my	per	the	up	you
myself	perhaps	their	upon	your
name	please	them	us	yours
namely	put	themselves	very	yourself
neither	rather	then	via	yourselves
never	re	thence	was	z

References

[1] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, O Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. UMass at TREC 2004: Novelty and HARD. In *Proceedings of the 13th Text Retrieval Conference*, TREC, 2004.

[2] Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval'12, pages 385–393, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[3] Eneko Agirre and Philip Glenny Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer, Dordrecht, The Netherlands, 2007.

[4] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and Novelty Detection at the Sentence Level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR'03, pages 314–321, New York, NY, USA, 2003. ACM.

[5] Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *J. Artif. Int. Res.*, 38(1):135–187, may 2010.

[6] Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. UNT: a Supervised Synergistic Approach to Semantic Text Similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval'12, pages 635–642, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[7] R Bar-Haim, I Dagan, B Dolan, L Ferro, D Giampiccolo, B Magnini, and I Szpektor. The Second PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

[8] M. Barathi and S. Valli. Ontology Based Query Expansion Using Word Sense Disambiguation. *IJCSIS*, 7(2):22–27, 2010.

[9] L. Bentivogli, P. Clark, I. Dagan, H. Dang, and D. Giampiccolo. The Seventh PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of TAC 2011 Workshop, NIST, Gaithersburg*, TAC '2011, 2011.

[10] Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa T. Dang, and Danilo Giampiccolo. The Sixth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of TAC 2010 Workshop*, TAC '2010, 2010.

[11] Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of Text Analysis Conference Workshop*, TAC'09, 2009.

[12] Carsten Brockmann and Mirella Lapata. Evaluating and combining approaches to selectional preference acquisition. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 27–34, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[13] Chris Buckley. New Retrieval Approaches Using SMART : TREC 4. pages 25–48.

[14] Aljoscha Burchardt, Marco Pennacchiotti, Stefan Thater, and Manfred Pinkal. Assessing the impact of frame semantics on textual entailment. *Nat. Lang. Eng.*, 15(4):527–550, oct 2009.

[15] Yee Seng Chan and Hwee Tou Ng. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th National Conference on Artifical Intelligence - Volume 3*, AAAI'05, pages 1037–1042. AAAI Press, 2005.

[16] Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. NUS-PT: exploiting parallel texts for word sense disambiguation in the English all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval'07, pages 253–256, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[17] Timothy Chklovski, Rada Mihalcea, Ted Pedersen, and Amruta Purandare. The SENSEVAL-3 Multilingual English-Hindi Lexical Sample Task. In *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, SENSEVAL-3, pages 5–8, Barcelona, Spain, 2004.

[18] Cortes Corinna and Vapnik Vladimir. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[19] Jim Cowie, Joe Guthrie, and Louise Guthrie. Lexical disambiguation using simulated annealing. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 157–161, Nantes, France, 1992.

[20] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question Answering Passage Retrieval Using Dependency Relations. In *Proceedings of the*

28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'05, pages 400–407, New York, NY, USA, 2005. ACM.

[21] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. *LNCS*, 3944:177–190, 2005.

[22] Hoa Trang Dang, Ching-yi Chia, Martha Palmer, and Fu-Dong Chiou. Simple Features for Chinese Word Sense Disambiguation. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING'02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[23] Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual, 2008.

[24] Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. *Identifying conflicting information in texts.* DARPA Global Autonomous Language Exploitation, 2012.

[25] Peter Wallis Dept and Peter Wallis. Information Retrieval based on Paraphrase. In *In Proceedings of PACLING Conference*, 1993.

[26] Takao Doi, Hirofumi Yamamoto, and Eiichiro Sumita. Example-Based Machine Translation Using Efficient Sentence Retrieval Based on Edit-Distance. *TALIP*, 4(4):377–399, 2005.

[27] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874, jun 2008.

[28] Ronald T. Fernández. *Improving Search Effectiveness in Sentence Retrieval and Novelty Detection.* PhD thesis, University Santiago de Compostela, 2011.

[29] Ronald T. Fernández and David E. Losada. Using opinion-based features to boost sentence retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM'09, pages 1617–1620, New York, NY, USA, 2009. ACM.

[30] Ronald T. Fernández and David E. Losada. Effective sentence retrieval based on query-independent evidence. *Inf. Process. Manage.*, 48(6):1203–1229, 2012.

[31] William B. Frakes and Ricardo A. Baeza-Yates. *Information Retrieval: Data Structures & Algorithms.* Prentice-Hall, Englewood Cliffs, N.J., USA, 1992.

[32] Norbert Fuhr. Probabilistic Models in Information Retrieval. *The Computer Journal*, 35:243–255, 1992.

[33] William A. Gale, Kenneth W. Church, and David Yarowsky. Using Bilingual Materials to Develop Word Sense Disambiguation Methods. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101–112, Montreal, Canada, 1992.

[34] William A. Gale, Kenneth W. Church, and David Yarowsky. Work on Statistical Methods for Word Sense Disambiguation. In *Working Notes, AAAI Fall Symposium Series, Probabilistic Approaches to Natural Language*, pages 54–60. AAAI Press, 1992.

[35] Michel Galley and Kathleen Mckeown. Improving word sense disambiguation in lexical chaining. In *In Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI)*, pages 1486–1488, Acapulco, Mexico, 2003.

[36] Miguel Angel Ríos Gaonac, Alexander Gelbukh, and Sivaji Bandyopadhyay. Recognizing Textual Entailment Using a Machine Learning Approach. In *Proceedings of the 9th Mexican international conference on Artificial intelligence conference on Advances in soft computing: Part II*, MICAI'10, pages 177–185, Berlin, Heidelberg, 2010. Springer-Verlag.

[37] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, 2007.

[38] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The Fourth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of TAC 2008 Workshop*, 2008.

[39] Daniel Gildea and Daniel Jurafsky. Automatic Labeling of Semantic Roles. *Comput. Linguist.*, 28(3):245–288, sep 2002.

[40] Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. Classification of semantic relations between nominals. *Lang. Resources and Evaluation*, 43(2):105–121, 2009.

[41] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan M. Cigarrán. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL'98 Workshop on Usage of WordNet for NLP*, pages 38–44, 1998.

[42] Cristian Grozea. Finding optimal parameter settings for high performance word sense disambiguation. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 125–128, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[43] Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. Negation, Contrast and Contradiction in Text Processing. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 755–762. AAAI Press, 2006.

[44] Donna Harman. Overview of the TREC 2002 Novelty Track. In *Proceedings of the Eleventh Text REtrieval Conference, NIST Special Publication 500-251*, TREC'02, pages 46–55, 2002.

[45] Adrian Iftene and Alexandra Balahur-Dobrescu. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE'07, pages 125–130, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[46] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding Semantically Similar Questions Based on Their Answers. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'05, pages 617–618, New York, NY, USA, 2005. ACM.

[47] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of the 14th ACM conference on Information and knowledge management*, CIKM'05, pages 84–90, New York, NY, USA, 2005. ACM.

[48] Valentin Jijkoun. Retrieving Answers from Frequently Asked Questions Pages on the Web. In *Proceedings of the 14th ACM conference on Information and knowledge management*, CIKM'05, pages 76–83, New York, NY, USA, 2005. ACM.

[49] Michael Kaisser. Answer Sentence Retrieval by Matching Dependency Paths Acquired from Question/Answer Sentence Pairs. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'12, pages 88–98, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[50] Abraham Kaplan. An experiment study of ambiguity and context. *Mechanical Translation*, 2(2):39–46, 1955.

[51] Marcin Kaszkiel and Justin Zobel. Passage retrieval revisited. *SIGIR Forum*, 31(SI):178–185, jul 1997.

[52] Marcin Kaszkiel and Justin Zobel. Effective ranking with arbitrary passages. *J. Am. Soc. Inf. Sci. Technol.*, 52(4):344–364, feb 2001.

[53] Robert Krovetz and W. Bruce Croft. Lexical Ambiguity and Information Retrieval. *ACM Trans. Inf. Syst.*, 10(2):115–141, apr 1992.

[54] Sadao Kurohashi. SENSEVAL-2 Japanese Translation Task. In *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Diasambiguating Systems*, pages 37–40, 2001.

[55] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 111–119, New York, NY, USA, 2001. ACM.

[56] Thomas K Landauer and Susan T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240, 1997.

[57] Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade. UMass at TREC 2002: Cross Language and Novelty Tracks. In *Proceedings of the 11th Text Retrieval Conference*, TREC, pages 721–732, 2002.

[58] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'01, pages 120–127, New York, NY, USA, 2001. ACM.

[59] Joon Ho Lee. Combining Multiple Evidence from Different Properties of Weighting Schemes. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'95, pages 180–188, New York, NY, USA, 1995. ACM.

[60] Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 41–48, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[61] Michael Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC'86, pages 24–26, New York, NY, USA, 1986. ACM.

[62] Xiaoyan Li. *Sentence Level Information Patterns for Novelty Detection.* PhD thesis, 2006. AAI3242322.

[63] Dekang Lin and Patrick Pantel. Concept discovery from text. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COL-

ING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[64] Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'04, pages 186–193, New York, NY, USA, 2004. ACM.

[65] David E. Losada. A study of statistical query expansion strategies for sentence retrieval. In *Proceedings SIGIR 2008 Workshop on Focused Retrieval (Question Answering, Passage Retrieval, Element Retrieval)*, SIGIR'08. ACM, 2008.

[66] David E. Losada. Statistical query expansion for sentence retrieval and its effects on weak and strong queries. *Inf. Retr.*, 13:485–506, 2010.

[67] David E. Losada and Ronald T. Fernández. Highly Frequent Terms and Sentence Retrieval. In *Proceedings of the 14th international conference on String processing and information retrieval*, SPIRE'07, pages 217–228, Berlin, Heidelberg, 2007. Springer-Verlag.

[68] Prodromos Malakasiotis and Ion Androutsopoulos. Learning textual entailment using SVMs and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE'07, pages 42–47, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[69] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[70] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[71] Daniel Marcu. The automatic construction of large-scale corpora for summarization research. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'99, pages 137–144, New York, NY, USA, 1999. ACM.

[72] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, LREC'06, pages 449–454, 2006.

[73] Marie-Catherine De Marneffe, Anna N. Rafferty, and Christopher D. Manning. Finding Contradictions in Text. In *Proceedings of Human Language Technology Conference*, HLT'08, pages 1039–1047, 2008.

[74] David Martínez, Eneko Agirre, and Lluís Màrquez. Syntactic features for high precision Word Sense Disambiguation. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING'02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

[75] Diana McCarthy and John Carroll. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Comput. Linguist.*, 29(4):639–654, December 2003.

[76] Charles T. Meadow. *Text Information Retrieval Systems*. Academic Press, Inc., Orlando, FL, USA, 1992.

[77] Rada Mihalcea. Co-training and Self-training for Word Sense Disambiguation. In Hwee Tou Ng and Ellen Riloff, editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 33–40, Boston, Massachusetts, USA, May 6 - May 7 2004. Association for Computational Linguistics.

[78] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 411–418, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[79] Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. The Senseval-3 English lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[80] Rada F. Mihalcea. Bootstrapping Large Sense Tagged Corpora. In *In Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC), Las Palmas*, 2002.

[81] George A. Miller. WordNet: a Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.

[82] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[83] Zhaoyan Ming, Tat-Seng Chua, and Gao Cong. Exploring Domain-specific Term Weight in Archived Question Search. In *Proceedings of the 19th ACM conference*

on Information and knowledge management, CIKM'10, pages 1605–1608, New York, NY, USA, 2010. ACM.

[84] Junta Mizuno, Eric Nichols, Yotaro Watanabe, and Kentaro Inui. Organizing Information on the Web through Agreement-Conflict Relation Classification. In *8th Asia Information Retrieval Societies Conference*, AIRS 2012, pages 126–137, 2012.

[85] Koji Murakami, Eric Nichols, Kentaro Inui, Junta Mizuno, Hayato Goto, Megumi Ohki, Suguru Matsuyoshi, and Yuji Matsumoto. Automatic classification of semantic relations between facts and opinions. In *The Second International Workshop on NLP Challenges in the Information Explosion Era*, NLPIX'10, pages 21–30, 2010.

[86] Koji Murakami, Eric Nichols, Junta Mizuno, Yotaro Watanabe, Shouko Masuda, Hayato Goto, Megumi Ohki, Chitose Sao, Suguru Matsuyoshi, Kentaro Inui, and Yuji Matsumoto. Statement Map: Reducing Web Information Credibility Noise through Opinion Classification. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND'10, pages 59–66, New York, NY, USA, 2010. ACM.

[87] Vanessa Murdock. *Aspects of sentence retrieval*. PhD thesis, University of Massachusetts Amherst, 2006.

[88] Roberto Navigli. Word Sense Disambiguation: A Survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, feb 2009.

[89] Hwee Tou Ng. Exemplar-Based Word Sense Disambiguation: Some Recent Improvements. *CoRR*, cmp-lg/9706010, 1997.

[90] Hwee Tou Ng. Does Word Sense Disambiguation Improve Information Retrieval? In *Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval*, ESAIR'11, pages 17–18, New York, NY, USA, 2011. ACM.

[91] Minh Hai Nguyen and Kiyoaki Shirai. Study on Supervised Learning of Vietnamese Word Sense Disambiguation Classifiers. *Journal of Natural Language Processing*, 19(1):25–50, 2012.

[92] Chikashi Nobata and Satoshi Sekine. Towards Automatic Acquisition of Patterns for Information Extraction. In *Proceedings of the 18th International Conference on Computer Processing of Oriental Languages*, ICCPOL'99, pages 11–16, 1999.

[93] Adrian Novischi. Combining Methods for Word Sense Disambiguation of WordNet Glosses. In *Proceedings of FLAIRS Conference*, pages 467–471. AAAI Press, 2004.

[94] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. SemEval-2010 Task: Japanese WSD. In *Proceedings of*, pages 69–74, 2010.

[95] Partha Pakray, Pinaki Bhaskar, Somnath Banerjee, Bidhan Chandra Pal, Sivaji Bandyopadhyay, and Alexander F. Gelbukh. A Hybrid Question Answering System based on Information Retrieval and Answer Validation. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[96] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[97] Jae Hyun Park and W. Bruce Croft. Query Term Ranking based on Dependency Parsing of Verbose Queries. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'10, pages 829–830, New York, NY, USA, 2010. ACM.

[98] Jae Hyun Park, W. Bruce Croft, and David A. Smith. A Quasi-Synchronous Dependence Model for Information Retrieval. In *Proceedings of the 20th ACM conference on Information and knowledge management*, CIKM'11, pages 17–26, New York, NY, USA, 2011. ACM.

[99] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th international conference on Computational linguistics and intelligent text processing*, CICLing'03, pages 241–257, Berlin, Heidelberg, 2003. Springer-Verlag.

[100] Thanh Phong Pham, Hwee Tou Ng, and Wee Sun Lee. Word Sense Disambiguation with Semi-Supervised Learning. In *Proceedings of the 20th National Conference on Artifical Intelligence (AAAI)*, pages 1093–1098. AAAI Press / The MIT Press, 2005.

[101] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[102] M. F. Porter. Readings in Information Retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[103] Amruta Purandare and Ted Pedersen. Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, USA, 2004.

[104] Dragomir Radev, Jahna Otterbacher, and Zhu Zhang. CST Bank: A Corpus for the Study of Cross-document Structural Relationships, 2004.

[105] Dragomir R. Radev. A Common Theory of Information Fusion from Multiple Text sources Step One: Cross-document structure. In *Proceedings of the 1st SIGdial workshop on Discourse and dialogue - Volume 10*, SIGDIAL'00, pages 74–83, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[106] Alan Ritter, Doug Downey, Stephen Soderland, and Oren Etzioni. It's a Contradiction—No, it's Not: A Case Study using Functional Relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'08, pages 11–20, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[107] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM'04, pages 42–49, New York, NY, USA, 2004. ACM.

[108] Stephen E. Robertson and Karen Sparck Jones. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[109] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. pages 109–126, 1996.

[110] Gerard M. Salton, Andrew Wong, and Chung Shu Yang. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620, nov 1975.

[111] Mark Sanderson. Word Sense Disambiguation and Information Retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'94, pages 142–151, 1994.

[112] Mark Sanderson. Retrieving With Good Sense. *Inf. Retr.*, 2(1):49–69, feb 2000.

[113] Hinrich Schütze and Jan O. Pedersen. Information Retrieval Based on Word Senses. In *Proceedings of the fourth annual symposium on Document Analysis and Information Retrieval*, 1995.

[114] Yohei Seki, David Kirk Evans, Lun-Wei Ku, Le Sun, Hsin-Hsi Chen, and Noriko Kando. Overview of Multilingual Opinion Analysis Task at NTCIR-7. In *Proceedings of NTCIR-7*, 2008.

[115] Kiyoaki Shirai. SENSEVAL-2 Japanese Dictionary Task. In *Proceedings of the SENSEVAL-2: Second International Workshop on Evaluating Word Sense Diasambiguating Systems*, pages 33–36, 2001.

[116] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33(1):6–12, 1999.

[117] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted Document Length Normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'96, pages 21–29, New York, NY, USA, 1996. ACM.

[118] Mark Stevenson and Yorick Wilks. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27:321–349, 2001.

[119] Christopher Stokoe, Michael P. Oakes, and John Tait. Word Sense Disambiguation in Information Retrieval Revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR'03, pages 159–166, New York, NY, USA, 2003. ACM.

[120] Carlo Strapparava, Alfio Gliozzo, and Claudiu Giuliano. Pattern abstraction and term similarity for Word Sense Disambiguation: IRST at Senseval-3. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[121] Michael Sussna. Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network. In *Proceedings of the second international conference on Information and knowledge management*, CIKM'93, pages 67–74, New York, NY, USA, 1993. ACM.

[122] Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL'06, pages 407–414, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[123] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[124] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In *Proceedings of the 13th annual international ACM SIGIR conference on Research*

and development in information retrieval, SIGIR'90, pages 1–24, New York, NY, USA, 1990. ACM.

[125] Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. Evaluating Variants of the Lesk Approach for Disambiguating Words. In *Proceedings of the ARPA Human Language Resources and Evaluation (LREC)*, pages 633–636, Lisbon, Portugal, 2004. European Language Resources Association.

[126] Ellen M. Voorhees. Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'93, pages 171–180, New York, NY, USA, 1993. ACM.

[127] Ellen M. Voorhees. Contradictions and Justifications: Extensions to the Textual Entailment Task. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, ACL'08, pages 63–71. Association for Computational Linguistics, 2008.

[128] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based QA Services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'09, pages 187–194, New York, NY, USA, 2009. ACM.

[129] Warren Weaver. *Machine Translation of Languages: Fourteen Essays (written in 1949, published in 1955).* MIT Press, 1955.

[130] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'96, pages 4–11, New York, NY, USA, 1996. ACM.

[131] Jinxi Xu and W. Bruce Croft. Improving the Effectiveness of Information Retrieval with Local Context Analysis. *ACM Trans. Inf. Syst.*, 18:79–112, 2000.

[132] David Yarowsky. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 266–271, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics.

[133] David Yarowsky. Decision Lists For Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[134] David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Nat. Lang. Eng.*, 8(4):293–310, 2002.

[135] Annie Zaenen, Lauri Karttunen, and Richard Crouch. Local textual inference: can it be defined or circumscribed? In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE'05, pages 31–36, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[136] Fabio massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. A machine learning approach to textual entailment recognition. *Nat. Lang. Eng.*, 15(4):551–582, oct 2009.

[137] Huaping Zhang, Hongbo Xu, Shou Bai, Bin Wang, and Xueqi Cheng. Experiments in TREC 2004 Novelty Track at CAS-ICT. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.

[138] Zhu Zhang, Sasha Blair-Goldensohn, and Dragomir R. Radev. Towards CST-enhanced summarization. In *Eighteenth national conference on Artificial intelligence*, pages 439–445, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[139] Zhu Zhang, Jahna Otterbacher, and Dragomir Radev. Learning cross-document structural relationships using boosting. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM'03, pages 124–130, New York, NY, USA, 2003. ACM.

[140] Zhu Zhang and Dragomir Radev. Combining labeled and unlabeled data for learning cross-document structural relationships. In *Proceedings of the First international joint conference on Natural Language Processing*, IJCNLP'04, pages 32–41, Berlin, Heidelberg, 2005. Springer-Verlag.

Publications

JOURNAL

[1] Minh Hai Nguyen and Kiyoaki Shirai "Study on Supervised Learning of Vietnamese Word Sense Disambiguation Classifiers," Journal of Natural Language Processing, vol.19, no.1, pp.25–50 (Mar. 2012).

INTERNATIONAL CONFERENCE

[2] Hai-Minh Nguyen and Kiyoaki Shirai: "A Study on Support Sentence Retrieval," Poster of the 8th International Conference on Natural Language Processing (JAPTAL 2012), (Oct. 2012).

[3] Hai-Minh Nguyen and Kiyoaiki Shirai: "Exploitation of Query Sentences using Specific Weighting in Support-Sentence Retrieval," in Proc.17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES-2013), (Sep. 2013).

[4] Hai-Minh Nguyen and Kiyoaiki Shirai: "Recognition of Agreement and Contradiction between Sentences in Support-Sentence Retrieval," in 8th International Conference on Knowledge, Information, and Creativity Support Systems (KICSS'2013), (Nov. 2013), *Best Student Paper Award*.