



THE HONG KONG
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library

包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

**A STUDY ON DISCOURSE TYPE BASED
INFORMATION RETRIEVAL**

WANG DAYU

Ph.D

The Hong Kong Polytechnic University

2014

The Hong Kong Polytechnic University
Department of Computing

**A Study on Discourse Type Based
Information Retrieval**

WANG Dayu

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

January 2009

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

_____ WANG Dayu _____ (Name of student)

Abstract

In ad hoc information retrieval (IR), some information need (e.g., find the advantages and disadvantages of smoking) requires the explicit identification of information related to the discourse type (e.g., advantages/disadvantages) as well as to the topic entity (e.g., smoking). Such information need is not uncommon and may not be easily satisfied by using conventional retrieval methods. So we propose the retrieval methods considering the discourse type of topics.

We propose IU similarity models and graph-based models to compute the similarity between a part of document (called information unit, IU in short) and a set of topic entity terms. Experimental results show that our IU similarity models with different term weighting schemes perform quite well and they are able to overcome the difficulties caused by the small size of IU. We also propose graph-based models which can compute the similarity of an IU based on topic entity terms only or based on both topic entity terms and discourse types based terms. In graph-based models, the basic unit is an edge that links two terms which are possibly two distinct topic entity terms, or a topic entity term and a discourse type term. These two models can be regarded as baselines of IU-based retrievals that do not rely on any discourse type information.

In actual documents, some individual terms are not adequate to present a discourse type. We focus on text patterns that have more powerful expression ability. We use word sequences, POS-tag sequences and the mix of both to match phrases and expression in order to find the text patterns that relate with a specific discourse type. These text patterns can also be selected by regarding the different types of sequences

as features in a pattern recognition application. These text patterns are used to quantify whether an IU contains the information on a specific discourse type.

For evaluation, we focused on some discourse types that can easily be identified in the TREC topics that are not satisfied very well using conventional retrieval models. We evaluated the discourse type based retrieval using our novel retrieval models and based on the text patterns mined by some selection conditions or learning algorithms. We showed that our concept of discourse type and corresponding solutions are able to enhance the retrieval effectiveness for the selected TREC topics.

Acknowledgements

First and foremost, I would like to thank my supervisor for his extensive experience and incredibly broad vision to this thesis work, as well as his consistent support and encouragement during my research at HK PolyU.

I would like to thank Prof. Kam-Fai Wong, my co-supervisor, and Prof Kui Lam Kwok for their great efforts and support to my PhD study.

I am thankful to my research group members for their ideas, comments, suggestions and help in my PhD study. They are Karen W. S. Wong, Jack H. C. Wu, Yinghao Li and Edward K. F. Dang.

I am thankful to Ms Miu Tai of General Office in our department and Ms Ada So and Ms May Chu of Research Office for their countless detailed work and assistance during my years of study.

Finally, I am deeply beholden to my parents WANG Hui and LIU Lijuan, to my wife ZHU Zhonghua for their immeasurable love and support. My parents make me recognize the value of knowledge and education. I thank my wife for her enduring affection and understanding. No words in any natural language would be sufficient to thank them for all they have done for me.

Table of Contents

Abstract.....	iv
Acknowledgements.....	vi
Table of Contents.....	vii
CHAPTER 1	1
INTRODUCTION	1
CHAPTER 2 LITERATURE REVIEW.....	13
2.1 Query Categorization.....	14
2.2 Question Classification in Question Answering.....	18
2.3 Conceptual Representation.....	21
2.4 Discourse analysis and discourse connectives	25
2.5 Opinion Mining.....	27
2.6 Context Window Techniques	31
2.7 Causal Knowledge Acquisition	33
2.8 Application of N-grams.....	35
Summary	36
CHAPTER 3 IU SIMILARITY MODELS.....	39
3.1 Fuzzy Model	43
3.1.1 Constant term weighting	45
3.1.2 Features based term weighting	49
3.1.3 Fuzzy set based similarity	53
3.2 Graph-based Model.....	56
3.2.1 IU Graph	56
3.2.2. Subgraph of an IU graph.....	60
3.2.3 Edges Selection Schemes.....	66
3.2.4 Relevant Evidence Score of an Edge	70
3.3 Experimental Results	77
3.3.1 Experiment Setup.....	77
3.3.2 Experimental Results of Fuzzy Model.....	83
3.3.3 Experimental Results of Graph-based Model	90
Summary	100
CHAPTER 4	102
DISCOURSE TYPE BASED INFORMATION RETRIEVAL	102
4.1 Introduction of Discourse Type.....	102
4.2 Discourse Type Based Retrieval by using Word Sequences	113
4.2.1 The distribution of word sequences in relevant documents	116
4.2.2 The distribution of word sequences in irrelevant documents.....	124
4.2.3 Distribution features of word sequences	133

4.2.4 Cross validation experiments based on word sequences.....	142
4.3 Discourse Type Based Retrieval by using POS Tag Sequences.....	146
4.3.1 The distribution of POS tag sequence in relevant documents.....	152
4.3.2 The distribution of POS tag sequence in irrelevant documents.....	157
4.3.3 Distribution features of POS tag sequences.....	161
4.3.4 Cross validation experiments based on POS tag sequence.....	165
4.4 Discourse Type Based Retrieval by using Word-POS Tag Sequences.....	169
4.4.1 The distribution of word-POS tag sequence in relevant documents.....	174
4.4.2 The distribution of word-POS tag sequence in irrelevant documents.....	180
4.4.3 Distribution features of word-POS tag sequence.....	184
4.4.4 Cross validation experiments based on word-POS Tag sequences.....	188
4.5 Comparison and Analysis of the Different Types of Linguistic Sequence and Features Evaluation.....	192
4.5.1 Analysis of the different types of linguistic sequences.....	192
4.5.2 Combinations of the different types of linguistic sequences and Feature Evaluation	201
Summary.....	206
CHAPTER 5.....	208
APPLICATION OF PATTERN RECOGNITION TECHNOLOGIES IN DISCOURSE TYPE	
BASED INFORMATION RETRIEVAL.....	208
5.1 Introduction.....	208
5.2 Feature Construction and Selection.....	210
5.2.1 Vector formulation.....	210
5.2.2 Feature selection.....	210
5.2.3 Feature space transformation.....	214
5.3 Classifier Selection and Comparison.....	219
5.3.1 Naïve Bayes classifier.....	219
5.3.2 Decision tree.....	222
5.3.3 Logistic regression model.....	223
5.3.4 Support vector machine.....	225
5.4 Experimental Results.....	228
Summary.....	237
CHAPTER 6.....	238
CONCLUSIONS.....	238
APPENDIX.....	243
A1. Retrospective retrieval experiments based on word sequence.....	243
A.2 Retrospective retrieval experiments based on POS tag sequences.....	262
A.3 Retrospective retrieval experiments based on word-POS tag sequence.....	270
A.4 Retrieval performance after adding the discourse type terms into the queries with a new version of search engine.....	277
REFERENCE.....	280

CHAPTER 1

INTRODUCTION

One of the difficulties of information retrieval (IR) is that the search engine can not know the user's information need accurately. IR system can search for many kinds of media, including text, image, sound and video etc; however, for the convenience of users, these users always represent their information need by text, and usually by some words only. For example, when we use Google or Yahoo, we often type some keywords only. The problem is how we exactly know the user's information need by these words.

The difficulty of clearly knowing information needs results from the diversity of information need. It is well acknowledged that the effectiveness of retrieval systems varies substantially from one query to another. This may be due to the diversity of user information need and a retrieval system cannot perform well for all the different kinds of queries. The diversity of user information need also implies that some information need is very complicated to express. Complex information needs may not be easily satisfied by common retrieval systems since they still deploy a relatively simple representation of the user information need. Potentially, the diversity of user information need may be one of the basic problems in IR.

The diversity lies in the variety of users and the complication of information. The

users are various in that they are different in their interests, their intention, and their background knowledge. The complication of information causes that even when the users want to know about the same interested topic, they probably focus on the different aspects or views of this interested topic. For example, a group of users want to get the relevant information on “cigar smoking” and what they really need may be one or of some of the following topics:

- popularity of smoking people in one country
- advantages and disadvantages of cigar smoking
- impact on economy or environment or healthy
- objection to cigar smoking from some organizations
- When to smoke hurts heart most?
- Does smoking cause obesity?
- Why smoking increases frequency of lung cancer?

Traditional IR model can not solve the problem of information diversity so well because the information need is only represented by a set of topic terms. For example, a user may describe the first topic by “popularity smoking people China”. In vector space model, a typical and classical model in traditional IR, the set of topic terms will be transferred into an n-dimension vector. In the evaluation of relevance of candidate documents, traditional IR techniques pay less attention to the different aspects (called “discourse type” by us) of information need of each query. For example, “popularity” is the discourse of query “popularity smoking people China” and it is probably represented by a concrete number in relevant document. The ignorance of different discourse types of queries causes that there are much difference in the performances of

using the same retrieval engine to retrieve relevant documents. We hold that considering the discourse type will understand the users' information need better and then improve the queries which have the poor performance.

We take some TREC topics as examples because it is generally accepted that these topics include a clear statement of what criteria make a document relevant. The format of a TREC Robust Track topic statement has been stable since TREC-5. A topic statement generally consists of four sections: an identifier, a title, a description and a narrative. The title field consists of up to five words that best describe the topic. The description field is a one sentence description of the topic area. The narrative gives a concise description of what makes a document relevant. Please look at the following TREC topic No. 308:

Table 1.1 TREC topic No. 308 "Implant Dentistry"

<num> Number: 308

<title> Implant Dentistry

<desc> Description:

What are the advantages and/or disadvantages of tooth implants?

<narr> Narrative:

A tooth replacement procedure, begun in the 1960s by Doctor Branemark, is becoming more widely used today. It involves the replacement of a lost tooth/teeth by an implantation process which secures the fabricated tooth to a titanium post with an adhesive resulting in a stable and sturdy denture almost like the original. A relevant document will include any clinical experiment, report, study, paper, or medical discussion which describes the advantages or disadvantages of tooth implant(s), conditions under which such a procedure is favorable, denture comfort and function compared to false teeth, bridge, or plate and comparative cost differential.

From the “title” part, we can find the entity related with this topic is “Implant Dentistry”, which is called “topic entity” in this thesis, and this topic entity is not adequate enough to completely describe the information need. From “description” part, we know more clearly about the information need, which is the “advantages and/or disadvantages” of the topic entity. We say that the discourse type of this information need (presented by the topic and further presented by a query to submit to the search engine) is “advantages and/or disadvantages”.

Let us look at how the relevant information exists in the relevant documents. We show two text passages extracted from the relevant documents of topic No.308.

But manufacturers insist their success rates are high. Dr. David Wacker of the Encino-based Core-Vent Corp. cites a 96% success rate in 623 implant patients after five years. More than 95% of 800 Interpore implants are still functional five years later, says George Smyth, president of the Interpore International in Irvine. (Note: Above passage is extracted from TREC DOCNO LA070489-0051, which is relevant document of Topic No.308)

...For no implant works as well as the original human version.

Although the vast majority of implants have improved the lives of their recipients, there are also failures - some of which leave patients with even more pain and disability than they had before.

(Note: Above passage is extracted from TREC DOCNO FT944-17268, which is relevant document of Topic No.605)

We find that the first passage contains not only the topic entity term “implant” but also the advantages or positive evaluation on dental implant. It uses “success rate” as a standard and presents some figures. The second passage also contains both the topic entity terms and the disadvantage or negative evaluation. It uses the comparison with the original human version and relies on the patients’ comparative feeling between

after use and before use. Therefore, these two passages are both relevant to topic 308.

We show the necessary information for the second passage to be relevant in Figure 1.1.

Figure 1 Analysis of discourse type based relevance.

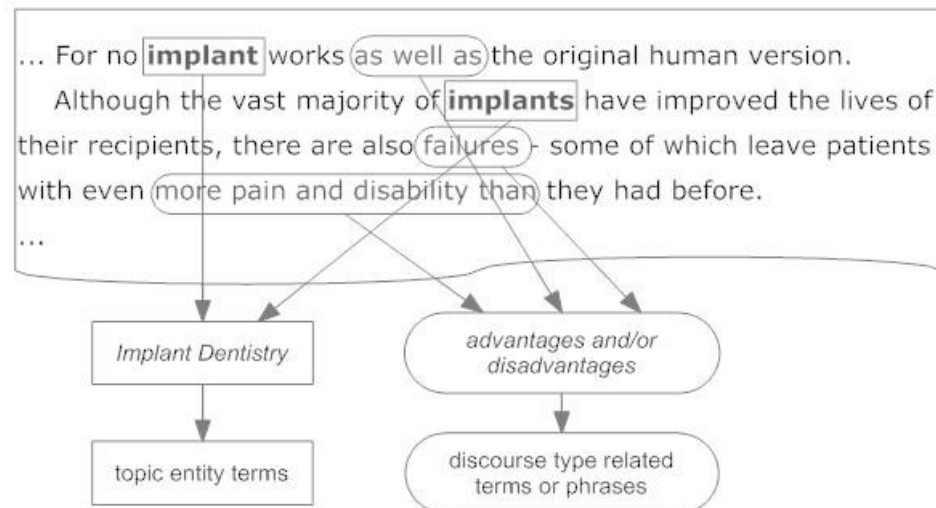


Table 1.2 TREC topic No. 605 “Great Britain health care”

<num> Number: 605

<title> Great Britain health care

<desc> Description:

What are the pros and cons of Great Britain's universal health care system?

<narr> Narrative:

Documents that discuss recommendations for change or list criticisms of the current system are relevant. Documents about an individual's experience with the health care system in Great Britain are irrelevant unless the document also contains a specific recommendation or criticism.

We know that, in most of time, only submit the topic entity terms to search engine cannot cover the whole information need. Even term expansion is ineffective since it only fetches the terms related with the topic entity. We also need the terms or phrases related with the discourse type. It's possible but not practical to provide a group of

discourse type related terms or phrase to support retrieval. We need some automatic and consistent methods for the same type of topics. If two topics have the same discourse type, can we discover similar text patterns in their relevant documents? We take another TREC topic as example:

It is not hard to find that the topic entity of topic No. 605 is “Great Britain's universal health” and the discourse type is “pros and cons”, which is very similar with advantages and disadvantages. Let us examine the relevant documents of this topic and we also present the following two passages as examples:

The number of patients treated in NHS hospitals has risen from 3.8m in 1951 to 9.2m in 1990-91. This has been achieved on a much smaller share of national income than most other advanced countries spend. UK health expenditure is just 6.5 per cent of GDP, compared with the average for the OECD of 9 per cent and for the EC of 7.6 per cent.

(Note: Above passage is extracted from TREC DOCNO FT923-6382, which is relevant document of Topic No.605)

After the first year's extra increases, we will restore the link between increases in the basic pension and prices or earnings, whichever is higher. Britain's national insurance system is far more efficient than private insurance.

(Note: Above passage is extracted from TREC DOCNO FT921-2350, which is relevant document of Topic No.605)

By the analysis shown in figure 1.1 we can find that above two passages also contain the discourse type related information as well as the topic entity terms. Also, we find that there are some common patterns to express “advantages/disadvantages” in these four passages, such as comparative adjectives and adverbs, comparison structures and numbers. This encourages us to mine these patterns for all the topics that have this

discourse type. (There are totally eight topics with this discourse type and we will introduce them in Chapter 4).

Obviously, for discourse type based retrieval, there are two factors that determine the relevance of a passage of text. One factor is to contain the information related with topic entity and the other is to contain the information related with the discourse type. In order to avoid scanning all text from the beginning to the end for the two kinds of information, we extract passages around the topic entity terms from documents and these passages are called information units (IU) in this thesis. We assume that all the IUs contain the information on topic entity and then we only need to check whether an IU contains the discourse type information.

Although an IU contains the information on topic entity, the relevance of different IUs with a topic entity may be quite different. In Chapter 3 we propose IU similarity models to quantify the similarity between an IU and a topic entity. Fuzzy models are based on individual topic entity terms and graph-based model are based on a pair of them. In fuzzy model, there are several weighting methods (constant, term feature based and fuzzy set based) for topic entity terms occurring in an IU. In graph-based models, an edge is linking two terms and there are many ways to select the edges for calculation.

Chapter 4 aims to solve the problem of how to determine whether an IU contains the information on a specific discourse type. The discourse type based retrieval models are based on linguistic sequences including word sequence, POS tag sequences and the mix of both. The discourse type based retrieval models can also evaluate these sequences. In order to select the most representative sequences, we propose some

measures to evaluate the sequences based on their distribution in the IUs of the retrieved relevant and irrelevant documents. These measures are aggregated into some sequence selection conditions and these conditions are evaluated in the retrospective experiments. The selected sequences can be taken into account directly in retrieval in our discourse type based retrieval models. Alternatively, the selected sequences can be regarded as features when we model the problem of determining whether an IU contains the information on a specific discourse type as a pattern recognition application, which is introduced in Chapter 5. All the 250 topics of TREC Robust Track 2004 are examined and grouped by their discourse types. We selected some discourse types containing adequate topics as examples. Experimental results of Chapter 3, 4, and 5 are based on these examples.

This thesis embodies several novel and significant contributions:

1. We propose a new approach to solve the problem of information diversity which can fundamentally enhance information retrieval. We put forward the concept of discourse type to accurately describe an information need in order to achieve better performance in retrieval by relying on the discourse type information. We manually examined all the TREC Robust Track topics and find some groups of topics with the same discourse type. We also select three discourse types that are significant in term of statistics and are appropriate for machine learning methods. We are able to justify the effectiveness of our methods on these three discourse types that are very abstract and not easy to cope with.
2. In order to study the discourse type based retrieval and simplify the procedure

of detecting the discourse type information, we divide a documents into some IUs(information unit, a special text window). We formulate IU-based retrieval which relies on IUs rather than the whole documents. We propose two IU similarity models (fuzzy model and graph-based model) to compute the similarity of IU with a query, which can be generalized into the similarity between a set of terms and a passage of text. Fuzzy model and graph-based model are used as baselines of not using discourse type information, which can be compared with the retrieval performance of using discourse type. We show that although the fuzzy model with constant weighting methods is set-based and very simple, it is comparable to the complex similarity models (e.g. vector space model, 2-possion model).

3. In order to discover the phrases or text patterns that can express a discourse type, we study the performances of word sequences (bigram, trigram, 4-gram), POS-tag sequences (bigram, trigram, 4-gram, 5-gram) and word-POS tag sequences (bigram, trigram) by using them to match the text in the IUs of the relevant documents of the topics with some specific discourse types. Some measures or features are proposed in order to select the representative sequences that are frequently and prevalently used for a specific discourse type. These measures sufficiently reflect the distribution of a linguistic sequence or wildcard sequence (may contain POS tag that corresponds to many different words) in relevant and irrelevant IUs, which conquers the difficulties of traditional statistics for “bag of words” methods. Experiments shows that these measures (e.g. QF n2, the ratio of the sum of query RDFs to

the sum of query IDFs) can help to discover the discourse type related sequences with very small occurrence frequencies. We innovatively show and deeply analyze the powerful ability of “pw” type (POS tag+word or word+POS tag) sequence to detect the discourse type information. We also compared our discourse type based retrieval model with the popular classifiers such as support vector machine and the results is our retrieval models are better than these popular classifiers in determine whether an IU is relevant to a topic with discourse type.

4. From the observations on the distribution of different types of sequences, we put forth a general method based on normalized Zipf’s curve which can reflect the quality of a linguistic sequence for detecting and presenting discourse types. This method is a good tool to evaluate a type of linguistic sequences, which is quite useful for the extension of our study.

Our research can be applied to any web information retrieval system. It is quite easy for the web user to explicitly indicate the discourse type (such as “reason” or “pros and cons”) in the query. For example, we can add command character “#” into the query input as “*reason# automobile recall*” which means the user needs the information on the reasons for automobile recall. Another way is to put the common discourse types into a drop-down list for web users to select. If the web user does not indicates the discourse type he/she wants, the returned list will be generated by the original methods. If the web user explicitly indicates the discourse type, the returned list will be retrieved or re-ranked as discourse type based retrieval.

There are possibly many methods to achieve a computationally efficient application of the discourse type based IR into an existent information retrieval systems. For example, we find that “pw” (POS tag+word) type sequence is the best linguistic sequence type to determine the discourse type of a text passage. If we make use of POS tag information in the text, we must pay attention on the time complexity of POS tagging algorithms. We know that when we use popular Viterbi algorithm [Viterbi 67] to do POS tagging and it has $O(N^2T)$ time complexity in which T is the length of the word sequence to be tagged. Therefore, it is infeasible to POS tag a large amount of documents during the retrieval. There are generally two methods to efficiently make use of discourse based information in actual application.

Let us take “pw” type sequence as an example. One method is to POS tag all the documents in the collection in advance and the information on “pw” sequence is stored just as common terms. An additional inverted document index on the “pw” sequence is then built which, for every specific “pw” sequence, contains the list of the documents that this “pw” sequence occurs. Therefore, we can compute how a document is related with a given discourse type by checking the occurrence of the discourse type related “pw” sequences which can be learned from previous training.

Another method is to re-rank or re-calculate the similarity scores of the top-ranked documents (or passages) in the retrieved list. In this way, only selected top-ranked documents are POS tagged. We can then compute how these top-ranked documents are related with a given discourse type by checking the occurrence of the discourse type related “pw” sequences. This method saves storage space and save time on the additional indexing but slower to react compared with the first one. The first method is

more suitable to relatively stable text collection such as legal documents and historic archives while the second method is more suitable to newly updated content such as reviews of a new movie or the instant feedback on an unexpected accident.

Our research can be extended by studying the other discourse types. Empirically, with the appearance of more TREC queries or other clearly-defined information need, more topics (queries) and relevant documents can provide us the possibility of improving and testing our study by exploring more discourse types.

Our research can also be extended by proposing other types linguistic sequences. In grammar, a part of speech (POS) is a linguistic category of words, which is generally defined by the syntactic or morphological behaviors of the words. Princeton's WordNet 3.0 version [WordNet 06] categories 155,287 words into 117,659 synsets as synonym sets. We use POS tag as element of a linguistic sequence and we know POS tag is a very coarse category. We think synset is so fine-grained for our study. So if there appear other systematic and reliable word categories, we can use it as element of linguistic sequences so that we can rely on the new linguistic sequences with different specificity. We can evaluate the ability of new linguistic sequences to detect or match discourse type information.

CHAPTER 2 LITERATURE REVIEW

Our work is to study a way to retrieval documents based on clearly knowing the information need by decreasing the diversity of information need. Comparably, query classification (categorization) is a coarse method to decrease the diversity of information need, which is reviewed in section 2.1. Section 2.2 introduces Answer Type Classification in Question Answering, which always deals with concrete information and simple abstract information. In section 2.3 we review some studies on some conceptual representations since they are related with our graph-based model. Our so-called “discourse type” makes people think of discourse analysis so it’s necessary for us to review some studies on discourse analysis and discourse connective in Section 2.4. We review the recent opinion mining studies in Section 2.5 because opinion mining is similar with the discourse type “advantage/disadvantage” we discover. In Chapter 3, we propose the concept of information unit (IU) in this thesis which is a fixed size text window extracted from the document and the centre term of the text window is one of the topic entity terms of the topic (query). So IU can be regarded as a context window around the topic entity term. So in Section 2.6 we review the context window concept of other studies to make a comparison. We review the studies on causal knowledge acquisition in the Section 2.7 because there is a discourse types that we deeply investigate in the following chapter: “reason”. In Chapter 4, we are using linguistic sequences of different lengths as patterns to match

the text. For example, a word 5-gram is a sequences consisting of five adjacent words. The concepts of bigram, trigram are not novel so we review some other studies which utilized the N-grams in Section 2.8.

2.1 Query Categorization

We review the studies on query categorization because their approaches have the same idea and purpose with our discourse types based information retrieval in that we all want solve the diversity of queries by classifying them. We use discourse type as categorization standard and they consider different features of queries such as geographical locality, time etc. The work which has attempted to solve the diversity of information need by classifying queries into different categories is called “query classification” or “query categorization”. The past query categorization methods can be grouped according to categorization standards such as intention, geographical locality, time and subject etc.

Query classification based on intention suggested that the users’ need differs according to their different intentions. [Broder 02] thought that the need behind a web search is often not informational -- it might be navigational (e.g. “give me the URL of the site I want to reach”) or transactional (e.g. “show me sites where I can perform a certain transaction, e.g. shop, download a file, or find a map”). They explore this taxonomy of web searches and discuss how global search engines evolved to deal with web-specific needs. They classified web queries according to users’ intent into 3 classes:

- Navigational: the immediate intent is to reach a particular site.
- Informational: the intent is to acquire some information assumed to be present on one or more web pages.
- Transactional: the intent is to perform some web-mediated activity.

[Kang 03] held that we need different strategies to find target documents according to a query type. They classified user queries as three similar categories with [Broder 02]: the topic relevance task, the homepage finding task, and the service finding task. [Azzopardi 06] held that acquiring the users' intentions is an important phase in the querying process and the identification of users' intentions enables the selection of appropriate retrieval strategies. They focused on one particular type of intention: the syntactic and semantic types associated with a query term and presented a case study using the email search task of the TREC Enterprise Track. They attempted to analyze the query to extract the meaning, semantics and nature of the query.

We show [Broder 02], [Kang 03] and [Azzopardi 06]'s studies in order to introduce how to categorize a query in the most general way: psychological intention behind the users. Obviously, most of the current text information retrieval studies, which includes our study, should be grouped into "Informational" category of [Broder 02], and "topic relevance task" category of [Kang 03].

Documents in traditional information retrieval are always independent and lack links to each other. However, web pages provide more information aside from content of text, such as page link information. Some people do query classification based on geographical locality and they suggested that information resources on the Internet are relevant to limited geographical communities. [Ding 00] proposed some strategies to

compute geographical scope of web pages based on geographical distribution of links to the page and geographical reference in the text of the page. [McCurley 01] presented a variety of approaches for recognizing geographical references on web page together with a navigational tool to browse pages by geographical proximity and their spatial context. [Jones 02] provided a brief survey of existing facilities for geographical information retrieval on the web. [Gravano 03] addressed this problem by first defining how to categorize queries according to their (often implicit) geographical locality. Then, they introduced several alternatives for automatically and efficiently categorizing queries in their scheme by machine learning techniques. They reported a thorough evaluation of their classifiers using a large sample of queries from a real web search engine, and conclude by discussing how query categorization approach can help improve query result quality.

In addition to geographical location, time is another important dimension of any events. Some people study on how queries change over time. Time analysis of queries is also an effective way to decrease the complexity of diverse queries. [Beitzel 04] focused on investigating the nature of changes in the query stream of a very large search service over time. They held that understanding how users' queries change over time is critical to developing effective, efficient search systems and to engineering representative test sets and evaluations that drive this development. They found trends over time are stable despite continuing fluctuation in query volume. Although the average query is repeated only twice during any given hour of the day, the total query traffic varies both in magnitude from one hour to the next, and also in degree of overlap and correlation in popularity of the queries that are received. In addition, they

also discovered that the frequency distribution of an hour's worth of queries remained constant throughout the day. Also, at the most general level, they found that query volume was highest and query sets were most stable during peak hours of the day.

The object in our study object is just the text of queries and documents regardless of the time when the text came into being and the geographical location where the writer wrote the text. We admit that the time and location may reflect the information need of a user, however, in this thesis we only study the information need presented by the text only.

Query categorization based on subject observed users' search interests by analyzing the subject or topical contents of the queries. [Pu 02] considered the problem of developing an automatic categorization method that is effective in classifying each term in the query into one or more appropriate categories that indicate the subject domain(s) of search interests. They constructed their subject taxonomy containing 15 major categories and 85 subcategories, including Adult, Arts & Humanities, Business & Finance, etc. Each major category consisted of several subcategories as well. Personalization of web search is to carry out retrieval for each user incorporating his/her interests. [Liu 02] proposed a novel technique to map a user query to a set of categories, which represent the user's search intention. This set of categories can serve as a context to disambiguate the words in the user's query.

Compared with the query categorization based on intention, time and geographical location, [Pu 02] and [Liu 02] are more similar with our study: to semantically categorize the queries. However, their methods are apparently easy to implement but are not effective. For most of the nouns occurring in the queries, it is

impossible to assign only one category to them. For example, “Clinton” may relate with foreign affairs, finance, litigation and humanities. It may refer to former US president and also his wife. Also, their studies lack of an evaluation mechanism. Our work is based on TREC queries and collection and we can evaluate easily and reasonably.

In conclusion, all the above methods of query categorization suggested that search engine should adopt different search strategies according to the different query categories. As for the standards, the intention, geographical locality, time are all background information on the user which are not related to the query content. The fourth standard we review is based on subject and this standard tries to discover the query content. This semantic consideration relates with our study. However, they just grouped the queries into some pre-defined categories rather than work out some methods to discover the accurate information need of a query. Also, they failed to offer a retrieval solution after recognizing a category.

By comparison, traditional IR models without query categorization assume that the documents in the collection are uniform and they retrieve documents according to the content of queries regardless of the categories of the queries. This might explain why traditional IR models perform differently depending on different discourse types in our experiment introduced in Chapter 4.

2.2 Question Classification in Question Answering

We review question classification in Question Answering (QA) system because the

idea to find the type of the answer is quite similar to our discourse type. Also, the topic presentation of TREC queries always contains a question in the “Description” section, for example, “What are the pros and cons of Great Britain's universal health care system?” of topic No.605. This question format is similar with the questions in QA systems. The difference is QA aims to discover an exact and accurate answer so the type/format and the content of answer (e.g. a date, a city or a reason) are very strict. Our task is to evaluate the relevance of a document to a topic and score the documents according to the relevance level rather than identify the most relevant document, sentence or even words.

Question Answering system attempts to retrieve correct answers to questions raised in natural language. The type of answer required is related to the content of the question, so knowing the type of a question can provide information on what relevant data is. Researchers in the field of Question Answering (QA) used question classification to analyze the question to a degree that allows determining the “type” of the answer. They have proposed various taxonomies for question classification. For example, [ISI 02] categorized 18,000 online questions with respect to their answer types. From this they derived a set of currently 115 elementary “Qtarget”s. [Li 02] defined a two-layered taxonomy to represent a natural semantic classification for typical answers in the TREC task. The hierarchy contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes.

[Moldovan 00] put forward a taxonomy of Question Answering Systems, in which the degree of complexity increases from Class 1 to Class 5. Class 1 depends on

dictionaries as knowledge base and use simple heuristic and pattern matching for reason, including questions like “Which is the largest city in Germany?” Class 2 uses ontology and low-level reason, including “How did Socrates die”. Class 3 is more complicated, including “What are the arguments for and against prayers in school.” Class 4 and 5 are even more complex. [Moldovan 00] classified 153 questions and as a result 136 questions (88.9%) belong to Class 1 and the rest 17 to Class 2. This example supports our claim above: most of the questions in QA require concrete information and simple abstract information.

Some machine learning methods has been used to solve the problem of question classification. [Suzuku 03] used a kernel function, Hierarchical Directed Acyclic Graph (HDAG) Kernel and they used 5011 Japanese questions that are labeled by 150 question types. [Blunsom 06] used a log-linear model and [Pan 2008] used a support vector machine model for question classification.

The concrete information extraction mostly depends on the semantic information of a single word or phrase while the abstract information depends on the relationships between words and phrases. For example, the advantages of an entity are generally presented by a complicated sentence rather than a few words. So, as the traditional vector space model, the previous studies in the QA field mentioned above have a tendency to depend on individual words rather than the relationships of words, although they used different mathematic models. Their methods are effective in finding concrete answers which are prevalent in QA field, but their methods are limited in finding abstract and complicated information.

In conclusion, these question classification methods in QA are based on the type

of answers. In this point, QA methods are similar to ours in that both methods concern with the type of information that users need. A user will express his need by a question for a QA system or by a query for retrieval system. However, most of the questions in QA require concrete information and very simple abstract information, whereas the ad hoc queries (e.g. TREC Robust track queries) are more complicated and diverse. The different types of information need in QA field and ad hoc information retrieval result in the different solutions. Compared with the popular methods used in QA fields, our methods based on query discourse type can investigate more complex relationships appearing in ad hoc queries.

2.3 Conceptual Representation

In Chapter 3, we propose a graph-based model to present a passage of text. So it is necessary to review some similar studies on conceptual representation which aim to present information in graphs. By comparison, our graph is not based on all the words occurring in the text but on some selected words. This way can effectively avoid the complexity of the graph and simplify the process of generating a graph.

Concept representation represents the information using a graph. After an information need is represented by such a graph, the entities and relationships among entities can be explicitly shown.

A conceptual graph (CG) is a notation for a system of logic based on the existential graphs of Charles Sanders Peirce and the semantic networks of artificial intelligence. At first, John F. Sowa used them to represent the conceptual schemas used

in database systems. Later, [Sowa 76], [Sowa 99] applied them to a wide range of topics in artificial intelligence, computer science, and cognitive science. A linear notation, called the Conceptual Graph Interchange Format (CGIF), has been standardized in the Final Committee Draft of the proposed ISO standard for Common Logic. CGs express meaning in a form that is logically precise, humanly readable, and computationally tractable. With a direct mapping to language, CGs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages. With their graphic representation, they serve as a readable, but formal design and specification language. CG's have been implemented in a variety of projects for information retrieval, database design, expert systems, and natural language processing. Apparently, [Sowa 76] and [Sowa 99] put forward an important concept and definition for graphic presentation of information. Empirically, their CG's highly depends on the parsing and semantic analysis which are both known to be quite hard to achieve high accuracy. This limits their application.

[Leskovec 04] presented a method for summarizing document by creating a semantic graph of the original document. The substructures of such a graph are identified to extract sentences for a document summary. They firstly did deep syntactic analysis of the text and, for each sentence, extracted logical form triples ““subject–predicate–object””. Then, they applied cross-sentence pronoun resolution, co-reference resolution, and semantic normalization to refine the set of triples and merged them into a semantic graph. [Leskovec 04] has the same problems with Sowa's work in that it also highly depends on the rules-based natural language processing which are limited by the performance and the efficiency.

[Niwa 97] generated a topic graph for each retrieved documents to make the users clearly know what a document talks about. First, topic words were extracted by the importance of words and by using frequency classes. Then, links are generated by co-occurrence analysis. Finally, a graph is mapping to a 2-dimensional area. The whole procedure is shown by the following figure in their paper. [Niwa 97] is a good tool to explicitly show the topic and content of a document to the IR users. They measured the importance of a word by its term frequency and this is a simple and effective ways. But their work does not intend to enhance the retrieval performance.

In the field of educational psychology, a concept map [Novak 90], [Novak 08] can be use to express complex ideas. It is a diagram showing the relationships between concepts. These concepts are connected with labeled arrows to construct a downward-branching hierarchical structure. The relationship between concepts is denoted by linking phrases such as "gives rise to", "results in", "is required by," or "contributes to". Concept maps have been used to stimulate the generation of ideas, and are believed to aid creativity. For example, concept mapping is sometimes used for brain-storming. Although they are often personalized and idiosyncratic, concept maps can be used to communicate complex ideas. Concept mapping can also be seen as a first step in ontology-building, and can also be used flexibly to represent formal argument. Actually, the relationships proposed by Novak's work are very representative and important to detect and determine the abstract and complicated information of text. They mainly focus on the framework of such a graph and our work in Chapter 4 provides some solutions on how to detect these relationships from text.

A mind map [Buzan 96] is a diagram used to represent words, ideas, or other

items linked to and arranged radially around a central key word or idea. Mind maps are used to generate, visualize, structure, and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing. The elements of a given mind map are arranged intuitively according to the importance of the concepts, and are classified into groupings or branches with the goal of representing semantic or other connections between portions of information. Compared with the concept map, the structure of a mind map is a similar radial, but is simplified by having one central key word. The concept of “one centre in the middle” is quite similar with our information unit because we all assume the importance of the centre term. But our graph based model in Chapter 3 does not have a centre term. Our graph is built for IR application so that we have different ways of defining which terms are important. Our graph is based on the occurrence of all the topic entity terms.

A topic map [Park 02] represents information using topics (representing any concept including people, countries, files, and events), associations (representing the relationships between topics), and occurrences (representing information resources relevant to a particular topic). Topic Maps are also similar to concept maps and mind maps in many respects, though only Topic Maps are standardized in this respect. In addition, topic map contains “occurrences” which provides the background information of associated topics. We believe that the background information can effectively benefit the topic presentation and we will justify it by our review in Section 2.6. The context-window techniques also intend to provide a certain amount of background information, which will be also reviewed in Section 2.6. In our graph based model, we do not use the background information because we find that only

using topic entity terms can already achieve comparable performance.

All above conceptual representations can represent text explicitly so that concepts in the text are more amenable to machine manipulation than text itself. However, it is extremely difficult to generate these conceptual representations correctly and consistently from raw text because the generation procedure depends on advanced text processing techniques and the availability of extensive knowledge bases. Instead, in our graph-based model we simplify the graphical representations of the term relationships so that these graphs can be generated easily and at the same time we try to make retrieval effective by using these graphs.

2.4 Discourse analysis and discourse connectives

We use the term “discourse type” to describe the concrete aspect of an information need, such as the reason of an accident or a disadvantage of a new policy. In order to distinguish “discourse type” in this thesis from the “discourse” used in semantics, we introduce some basic background of discourse analysis and how this field relates to our work.

“Discourse” is a term used in semantics and discourse analysis. In semantics, discourses are linguistic units composed of several sentences — in other words, conversations, arguments or speeches. We use the term “discourse type” since the linguistic unit in our study, called information unit, is larger than a single sentence and it contains the relation between sentences. Hence we review some studies in the field of discourse analysis.

In M. Stubbs' textbook [Stubbs 83], discourse analysis is defined as (a) concerned with language use beyond the boundaries of a sentence/utterance, (b) concerned with the interrelationships between language and society and (c) as concerned with the interactive or dialogic properties of everyday communication.

Among above three branches, (a) relates with our discourse type based information retrieval in that the boundaries between the adjacent sentences can help us to find discourse relations (also called coherence relations, rhetorical relations, rhetorical predicates and conjunctive relations). At the boundaries of some sentences, there perhaps exist some discourse connectives, which are function words or phrases that signalling relations in discourse, such “however”, “for instance”, “as a result”, etc. Study of discourse relations contains [Martin 92]; [Knott 96], [Knott 98] and [Webber 99]. [Hutchinson 05] classified the discourse connectives, which are the words such as conjunctions at the beginning of a sentence. They intended to discover some typical discourse connectives for a specific relationship such as reason. We think their methods are able to achieve a high precision because the sentence after a causal connective (such as “because”) is very likely to be reason but their methods cannot be so effective for IR application which also pay attention on recall. They cannot find the implicit reason without the apparent connective very well, which we will give an introduction in Section 2.7 Causal Knowledge Acquisition.

The early work of [Brooks 83] introduced discourse analysis into IR. After that people in the field of computational linguistics pay more attention to the discourse analysis, one example is [Webber 03]. Some researchers have studied the discourse terms and discourse types. Work of [Knott 96] investigated how cue phrases determine

the coherence relations and work of [Hutchinson 05] studied the acquisition of discourse connectives. In our study, the discourse terms of some certain discourse types are similar with the cue phrases in [Knott 96] and discourse connectives in [Hutchinson 05]. However, many discourse types in our study are more implicit and complicated than the relations studies in discourse analysis. A lot of discourse connectives are discovered by statistical methods. Notwithstanding this fact, we can still make use of the relations defined and studied in discourse analysis.

2.5 Opinion Mining

We review the recent opinion mining studies because opinion mining is similar with several discourse types we discovered. For example, there is one discourse type “argument for and against” which is one way to evaluate. Also, there is another discourse type that we deeply studied: advantage/disadvantage, which is a common way to express opinions. Technically, they mostly depend on detecting whether any of the pre-defined words occur in the target text. Some of the work depends on word sequences. We innovatively use linguistic sequences consisting of different combination of POS tags and words to detect advantage/disadvantages.

Opinion mining, which is also called *sentiment classification* or *sentiment analysis*, is to classify opinion text into positive or negative evaluation of a target project (film, book, product etc.). Generally speaking, there are three main approaches to deal with the opinion mining.

The first approach is based on “bag of words”, which attempted to learn a

positive/negative document classifier based on occurrence frequencies of the various words, bigrams, trigrams in the document. For example, [Pang 02] used words and bigram as features without the help of stemmer or stoplist. They used several supervised machine learning methods (naive Bayes, maximum entropy and SVM) to do sentiment classification on movie reviews from Internet Movie Database (IMDB). [Dave 03] used unigram, bigram and trigram as features and also experiments a number of learning methods to classify the reviews on computer & consumer electronics products.

The second approach is “semantic orientation”, which assigned words scores indicating “positive” and “negative” and then aggregated the word scores into an overall score for the whole text. [Turney 02] used two-word phrases that contain adjectives or adverbs as features. They applied an unsupervised learning technique based on semantic orientation, which is equal to the difference of mutual information between these phrases and the words that indicate “excellent” and “poor”. They classified the reviews on automobiles, banks, movies and travel destinations.

The following papers also discuss how to compute semantic orientation although they did not put semantic orientation into opinion mining task. [Hatzivassiloglou 97] computed semantic orientation of adjectives, assuming that conjunction “*and*” usually conjoins two adjectives of the same orientation, while “*but*” conjoins two adjectives of opposite orientation. [Turney 03] determined the semantic orientation of terms by bootstrapping from a pair of two minimal sets of “seed” terms:

positive set = {*good, nice, excellent, positive, fortunate, ...*}

negative set = {*bad, nasty, poor, negative, unfortunate, ...*}

[Kamps 04] used short distance in WordNet between two terms to compute the semantic orientation. [Esuli 05] presented a method to compute the sentimental orientation of subjective term based on the quantitative analysis of the gloss of the terms. Gloss is the definition of term in the on-line dictionary. We think these two types of approaches are able to achieve a high precision for the explicitly presented discourse types but their methods cannot be so effective for achieving high recall. They cannot find the implicit reason without the apparent connective very well, which we will give an introduction in Section 2.7.

Unlike above two methods based on individual words, the third approach is based on complex semantic knowledge. They suggested that opinion mining needs detailed semantic analysis of attitude expression based on a well-designed taxonomy of attitude types and other semantic properties. [Martin 07] put forward “appraisal theory” and built a taxonomy called “Appraisal Groups”, which consists of these properties: ATTITUDE, GRADUATION, ORIENTATION, and POLARITY. [Whitelaw 05] used “Appraisal Groups” for opinion mining. They applied some semi-automated methods to build a lexicon of appraising adjectives and their modifier (e.g. *very*) and opinion mining was performed on movie reviews using features based on taxonomies. We think the approach based on complex semantic knowledge is theoretically sound and facilitates formulations of some existent methods. However, this approach highly relies on the parsing and semantic analysis which limits their feasibility and performance for IR applications.

Among the several discourse types we determined for the TREC queries, the discourse type that is most relevant to opinion mining is “argument for and against”,

because opinion mining is dealing with the positive or negative opinions in text and the relevant documents in response to queries whose discourse type is “argument for and against” may also contain people’s positive or negative opinions. However, the purposes of opinion mining and our study are different. For opinion mining, it’s known beforehand that the document contains opinion and the task is to determine the nature of opinion: positive or negative. As for opinion mining on reviews, it even assumes that the target object evaluated in the text is known. While in our study--- to determine the relevance of documents retrieved in respond to the queries whose discourse type is “argument for and against”, we need to confirm two things, one is whether a passage of document is talking about the target object and the other is whether this passage contains positive or negative opinions. Yet our work on the discourse type “argument for and against” is related with opinion mining in that both probably depend on the same subjective terms or phrases. In the same way, the discourse type “objection” relates with the recognition of negative opinion. There are some discourse types that may use people’s opinion to express, such as “advantage and disadvantage”. These discourse types are more or less related with opinion mining. Empirically speaking, the first two types of approaches are able to achieve a high precision for the explicitly presented discourse types but their methods cannot be so effective for achieving high recall. The third type of approaches which rely on complex semantic knowledge are easily affected by the parsing and semantic analysis which limits their feasibility and performance for IR applications.

2.6 Context Window Techniques

We propose the concept of information unit (IU) in this thesis which is a fixed size text window extracted from the document and the centre term of the text window is one of the topic entity terms of the topic (query). So IU can be regarded as a context window around the topic entity term. We review the context window concept of other studies to make a comparison.

It is well acknowledged that the context of a term provides related information of this term. This information can be used to disambiguate the terms with multiple senses and help to clearly understand the whole sentence and paragraph. Researchers in the field of information retrieval and computational linguistics have noticed this phenomenon and use the context information for different tasks. Their work is related with the concept of information unit proposed in this thesis.

Context windows are used in the task of finding collocates of given terms. “PhraseFinder” technique developed by [Jing 94] used context window to define collocates for automatically constructing a co-occurrence thesaurus. Each indexing unit has been stored in the thesaurus with a list of its most strongly associated collocates. Collocates are defined as index units co-occurring in windows of 3–10 sentences, which approximate the size of an average paragraph. The widely known Local Context Analysis proposed by [Xu 00] defined collocates of query terms as noun groups that are taken from the retrieved N top ranked passages of fixed size of 300 words. [Vechtomova 03] defined collocates of a single instance of the term as all words that occur within a fixed-length window surrounding this term. Each window is centred around a node term. In their work, a window is defined for each instance of

each query term in a set of relevant documents (local analysis) or in the entire collection (global analysis). In this point, their text window is very similar with our information unit.

There also appeared some studies on IR model based on term context. [Pickens 06] proposed a term context model, which assesses the presence of a term in a document based not on the actual observed occurrence of that term, but on the evidence of a set of supporting terms, or context. They have shown that their model is useful for retrieval in that it can improve the precision at low recall.

[Wu 05] proposed a novel model to compute the term weight for each of the matched query terms in the document of the based on the context information. The term weight is calculated by multiplying probabilities similar to the well-known probabilistic models (e.g. binary independence model) and language model. Their experimental results showed that that context information is important for information retrieval. [Wu 07] proposed a qualitative model of the process of making human judgment based on combining the local relevance decisions, which is determined by the information in the context around a core term. [Wu 08] further developed theoretical basis for above relevance decisions making. Our IU has very similar definition with Wu's work and the size of our IU is determined by the experimental results of Wu's [Wu 05], [Wu 07] and [Wu 08] because our baseline retrieval lists are generated by the same retrieval model on the same text collection.

2.7 Causal Knowledge Acquisition

Researchers have proposed different sets of semantic relations. CAUSATION is undoubtedly one of the most important relations. We review the studies on causal knowledge acquisition in this section because there is a discourse types that we deeply investigate in the following chapter: “reason”. The topics (queries) categorized into “reason” are looking for the reason of an event. For example, the “Narrative” part of TREC topic 673 states “Documents must provide a reason for the withdrawal (of Soviet troops from Afghanistan)”. Therefore, the relevance to the topics of discourse type “reason” is quite related with causal knowledge acquisition in that both are searching for the reason.

Many studies make use of cue phrases. [Girju 02] proposed a method to acquire causal knowledge from English text based on the triplet patterns

$$\langle _NP1 \textit{ clue} NP2 _ \rangle$$

where *clue* is a causative verb, and *NP1* and *NP2* are noun phrases. Causative verbs express a causal relation between the subject and object (or prepositional phrase of the verb), such as “cause” and “force.” They screen the causative verbs by semantic categories defined in WordNet. They manually evaluated 300 of 1300 patterns they extracted and the accuracy is about 65%.

[Girju 02] studied the relation between two noun phrases and [Marcu 02] worked on the relation between two sentences. To detect the causal relation from other rhetorical relations, they used the sentence pairs connected with “Because of” and “Thus”. Naïve Bayes classifier is used to classify the sentence pairs into either “causal” or not. The accuracy is about for inter-sentence causality extraction.

Through a lot of experiments and analysis, we find that the problem of depending on cue phrase is that causation is always expressed implicitly: it is not necessary for causation to exist with a casual marker. A causal marker is a linguistic unit signaling causal relation. The markers can be causal connectives: prepositional (such as because of, thanks to, due to), adverbial (such as for this reason, the result that), or clause links (such as because, since, for). The markers can also be causation verbs. However, causation can be expressed without these markers. Moreover, one of the difficulties of natural language processing is ambiguity. For example, “since” sometimes lead a causal clause and sometimes not. Sometimes, causation can be implicitly impressed when the effect, one of the arguments of causation, is not mentioned. Therefore, most of the past studies focus on the extraction of causation which is explicitly expressed with the markers.

Obviously, these methods (for example, one simple way is depending on connective “because”) has a high precision but low recall. However, in our application, relevance is determined if any part of a documents or any IU contains the required information. As for the presentation of the results of IR, the retrieved documents are ranked by a score rather than assigned a binary result of relevant or irrelevant. Therefore, any specific matching (high precision but low recall) is insufficient for IR applications. So we propose the different types of linguistic sequences with different specialties in Chapter 4. By using them, we can quantitatively control the specialties of the matching patterns to a required extent which is a feasible way to increase the recall.

2.8 Application of N-grams

In Chapter 4, we are using linguistic sequences of different lengths (such as “POS tag, POS tag, word”) as patterns to match the text. With the length of a linguistic sequence increases, the specialty increases. For example, a word 5-gram (or quintgram in Latin terminology) is a sequences consisting of five adjacent words. The concepts of bigram and trigram are not novel so we review some other studies which utilized the N-grams.

N-gram models are probabilistic models for predicting the next item in a sequence. N-grams are used in various areas of statistical natural language processing and genetic sequence analysis. An n-gram is a sub-sequence of n items from a given sequence. The items in question can be phonemes, syllables, letters, words or base pairs according to the application. N-grams are widely used in language model [Pont 98] of information retrieval and POS tagging based on Hidden Markov Model in natural language processing.

Some work in the field of opinion mining used word n-gram, POS n-gram and mix of word and POS tag n-gram as features. For example, [Turney 02] used some POS tag bigram patterns (see Table 1 in his paper) to extract two consecutive words from the reviews. For example, he used “JJ + NN (or NNS)” to extract the word bigrams that consist of an adjective followed by a noun and he used “RB, RBR or RBS + JJ” to extract the word bigrams that consists of an adverb followed by an adjective. [Dave 03] used word unigram, bigrams and trigrams as features to do semantic classification on product reviews. [Li 08] used mix of word and POS N-grams ($N \leq 3$) as features to determine the subjectivity orientation of questions posted by real users in community question answering (CQA) portals. They observed a slight gain with more

complicated features, e.g. word and POS n-gram.

The above studies used an n-gram as a general pattern to match the text. They have some expressions in advance. For example, if they want to find a noun phrase consisting of a noun followed by a modifying adjective, they use “JJ+NN” or “JJ+NNS”. It is a good way to find a noun phrase in this format but it cannot cover all the expression with the same function. Due to the complexity of natural language, the modifier of a noun can also be an adjective phrases and adjective clauses. It is almost impossible to matching an adjective clause by one or some patterns. For example, “the coat which I bought yesterday” includes an relative clause “which I bought yesterday” to modify “coat”. It is quite hard to match the whole clause by a POS tag N-gram. This one is possible to be matched by “NN+which” which is used in Chapter 4 as a “pw” type. We will explain our methods in details in the following chapter.

Summary

Strategically, we intend to solve diversity problem of information need in order to better satisfy a query and enhance the retrieval performance. In this point, our study is similar with query categorization because query categorization suggests that search engine should adopt different search strategies according to the different query categories. Query categorization can be based on different standards. The intention, geographical locality, time are all background information on the user which are not related to the query content. The fourth standard we review is based on subject and this standard tries to discover the query content, however, it just grouped the queries into

some pre-defined categories rather than work out some methods to discover the accurate information need of a query and provide the solutions.

We are using TREC documents and topics in this thesis. We manually analyzed the topics and grouped them into different discourse types. We review question classification in Question Answering (QA) system because the idea to find the type of the answer is quite similar to our discourse type. The difference is QA aims to discover an exact and accurate answer so the type of answer (e.g. a date, a city or a reason) is very strict. Our task is to evaluate the relevance of a document to a topic with clearly stated discourse type. Moreover, we intend to find very abstract information such as “advantage/disadvantage”, “reason”, while QA usually focus on very concrete information.

In details, we investigate three discourse types “advantage/disadvantages”, “reason” and “impact”. The discourse type “advantage/disadvantages” is similar with the studies reviewed in Section 2.5 opinion mining. Technically, they mostly depend on detecting whether any of the pre-defined words occur in the target text. Some of the work depends on word sequences. We innovatively use linguistic sequences consisting of different combinations of POS tags and words to detect advantage/disadvantages. The discourse type “reason” and “impact” are similar with the studies reviewed Section 2.7 Causal Knowledge Acquisition. They mostly depended on pre-defined cue phrases and this lead to very low recall since a lot of causal relations are not presented by causal verbs and connectives. Our linguistic sequences including different combinations of POS tags and words were proved as effective patterns to detect casual knowledge.

Methodologically, we used IU, the fixed size text window extracted from the document, as the basic unit so we review some studies on context window techniques. We use a simple graph to present the relationships between the topic entity terms in an IU so we review some studies on conceptual representation. We are using linguistic sequences possibly consisting of words and POS tags so we review some studies on N-gram. We use the term “discourse type” so we review some studies on “discourse analysis” to clearly explain our terminology.

CHAPTER 3 IU SIMILARITY MODELS

In this chapter, we propose IU similarity models to quantify the similarity between an Information Unit (IU), a part of document, and a topic, based on the single topic entity term (fuzzy model) or a pair of them (graph-based model). The purpose is to investigate the performance of IU-based retrieval without considering the discourse type information and build baselines for later comparison and analysis. Also, we intend to show that applying simple weighting schemes in IU-based retrieval can outperform the complicated weighting schemes and even the most popular weighting schemes (e.g. idf value). We have shown that our baseline is comparable to the document-based retrieval so outperforming the baseline is more promising.

We all know that a document is a big language unit so that a document may contain several topics which are presented by different topic entities, different properties of these topic entities and relationships among the topic entities. The judgment of the relevance of document adopted by TREC is based on the disjunctive relevance decision (DRD) principle [Kwong 04] which states that any part of a document that is considered relevant implies that the whole document is relevant. So we first decompose a document into small parts so that we can measure the relevance of each small part of a document to the topic.

[Wu 07] proposed a novel retrieval model to simulate human relevance decision making and their model explicitly models a human relevance decision at each

location in a document. The relevance decision at the specified location in the document is based on the context at that location so that the relevance decision preference (denoted by a relevance score) at the specified location is estimated using the context at that location. Their work also inspires us to decompose the document into small parts and to combine the scores of each part into a final score.

If one part of document is relevant to the required topic entity and this part also has the required discourse type, the document is relevant. There are many possible parts that can be extracted from a document. Obviously it's more efficient to process the parts that are very likely to cover the topic entity than process all possible parts. Hence, we put forward the concept "information unit".

For a document and a topic, Information unit (IU) is a fixed size text window extracted from the document and the centre term of the text window is one of the topic entity terms of the topic. We are studying TREC Robust Track topics and the topic entity terms are selected from the terms in the title section. We are doing this because the title section contains terms that best describe the information need and it's possible to compare our work with other's work based on title section. For example, in query "*term limits*" (title section of topic 699), "*term*" and "*limits*" are topic entity terms. For a document and a topic, an IU extracted from the document is a part of the document that contains at least one topic entity term in the centre. Apart from the topic entity term in the centre, the IU may contain other topic entity terms with different term frequencies at different positions. Also, the IU may contain some expressions to indicate the discourse type. In order to quantify how an IU is related with a given topic and we propose two IU similarity models to quantify the similarity. They are fuzzy

model and graph-based model.

Let T be a set of terms that describe the topic entities mentioned in a query. The terms in T are terms appearing in title section of the TREC Robust Track topics. We name the terms that belongs to T “topic entity terms”. Given a document doc , an information unit (IU) is defined as a fixed size text window that contains odd number (viz. $2w+1$) words and the word in the center of this text window belongs to T . So if an IU contains $2w+1$ words, we say that the size of this IU is $2w+1$. Obviously, for an IU whose size is $2w+1$, the $(w+1)$ -th word, the word in the center, of this IU belongs to T . If a query term appears in the first w words of a document, the first $2w+1$ words of this document is extracted as an IU. And in this case the centre term of an IU is not one of the topic entity terms. However, this case is very rare. Let us assume that the discourse type of query q is related with a set of discourse terms denoted by D . We also assume that a sliding widow of a document is relevant if and only if it’s an IU. Hence, relevance, measured by the similarity score, of a document doc only depends on the relevance of all the IUs in the document:

$$sim(doc, q) = Agg_1[sim(IU_i(doc), q)] \quad (3.1)$$

where $IU_i(doc)$ denotes the i -th IU in document doc , the $sim()$ is the similarity between the IU and q , and Agg_1 is an aggregation function.

We have noted that [Wu 08] defined the document-wide relevance as:

$$R_{d,q} \equiv C(\{R_{d,k,q} : k \in [1, |d|], k \in N\})$$

In above equation, $R_{d,q}$ denotes the document-wide relevance variable for document d and query q , $R_{d,k,q}$ denotes the local relevance variable at location k of document d for query q , $C(.)$ denotes the generic function that combines the outcomes of local

relevance decisions. More specifically, the document-wide relevance-decision function (denoted by $\nabla(d, q)$) for document d and query q is derived by combining all the outcomes from local relevance-decision functions at different locations in document d for query q (denoted by $\partial_{d,k}(c(d, k, n), q)$):

$$\nabla(d, q) = C(\{\partial_{d,k}(c(d, k, n), q) : k \in [1, |d|], k \in Z\})$$

The formula 3.1 is a special form of [Wu 08]’s definition and function described above in that (1) both formulate the document-wide relevance measure as the aggregation of the local relevance measures. $C(.)$ and agg_I both function as the aggregating or combining functions; and (2) [Wu 08] provided a more general formulation since it considered all the possible locations in a document. However, formula 3.1 only considers the locations where IU can be extracted, in other words, the locations that topic entity terms occur in the documents.

The choice of the function depends how to define the relevance of a document. For example, if the relevance of a document is determined by the most relevant IU in it, maximum is a good choice for Agg_I . If the relevance of a document is determined by the total amount of relevant information it contains, we can use summation to derive Agg_I . In our experiments, we use summation because it behaves like disjunctive function: any component with large value can substantially increase the sum and any component with small value cannot greatly decrease the sum. Also, experimental results show that using summation is better than using maximum or mean.

The similarity score $sim(IU_i(doc), q)$ in formula (3.1) can be computed differently using different similarity models defined over $IU_i(doc)$ and a set of terms. Since our later discussion focuses on an individual IU, we will drop the suffix I from

IU for the sake of brevity. The rest of this chapter is organized as follows. In section 3.1, we will introduce fuzzy model, in which some set-based similarity measures are adopted rather than the traditional vector-based measures. In section 3.2, we will introduce graph-based model which considers a pair of terms as the basic element in calculation of the IU similarity. In section 3.3, we report the experimental results of fuzzy models and graph-based models.

3.1 Fuzzy Model

The purpose of formulating a fuzzy model is to quantify the similarity between an IU and a set of distinct terms. It can be generalized into some similarity measures between two groups of terms. In fuzzy model, we use some set-based similarity measures rather than traditional vector-based measures because we use IU rather than the whole document as the basic unit. (It's because we assume that Ius are only relevant parts of a document when we derive formula (3.1)). Also, we assume that the relevance of an IU merely depends on the topic term set T and the discourse term set D . Since term frequencies in an IU are usually one due to the fact that IU has a much smaller size than a document. Simple set-based measures adopted in our fuzzy model are adequate, which is also justified by the experimental results. Second, vector-based model will equally score the repeated terms that occur in different positions in a document. In our fuzzy model, we can deal with them differently, for example, one measure is considering their distance to the center of IU. We cannot directly use traditional set similarity measures (such as Jaccard, Dice) because set theory assumes that elements

in a set are distinct but terms often repeat in text. We need to propose the similarity measures considering the repeated words.

Let S be a set of terms and IU be a sequence of terms that are in an IU. There are no repeated terms in S and there are possibly repeated terms in IU . The order of terms in S does not make any sense but the order of terms in IU makes sense. Let t_i be the i -th term of S . The suffix I in t_i is to distinguish the different terms. We use t_{ij} to denote term t_i that occurs at the j -th term position in IU . So when use t_{ij} , it's possible for two terms that has the same first suffix, such as t_{ij} and t_{ik} , which respectively refer to term t_i that occur at the j -th term position and k -th term position in IU . However, the second suffix in t_{ij} , viz. j , cannot repeat since it represents the term position in IU .

The general similarity function for an IU is

$$sim(IU, S) = f_{t_i \in S \wedge t_{ij} \in IU} (weight(t_{ij})) \quad (3.2)$$

where $f()$ is a aggregation function which can be specialized by Boolean operations, algebra operations etc. and $weight()$ denotes the weight of term t_{ij} . Formula 3.2 depends on whether and how the terms that belong to S occur in IU . We also propose three types of different term weighting schemes. In subsection 3.1.1, we introduce five constant term weighting schemes that weight all topic entity terms in the same way. These five term weighting schemes produce five similarity measures. In subsection 3.1.2, three feature-based term weighting schemes are proposed, which assign different weights to different terms based on the features (such as specialty, position etc.). In subsection 3.1.3, Dombi intersection operator is used as an example to introduce fuzzy set based term weighting scheme.

3.1.1 Constant term weighting

In this section, we introduce constant term weighting, in which all terms have equal weights. Apparently, this is the simplest weighting scheme and we use it as benchmark. Some of the constant weighting schemes can obtain fairly good results compared with complex weighting schemes introduced in later sections.

Given a term $t_i \in S$, simple constant weighting for term t_i is:

$$cw(t_i) = \begin{cases} 1, & \exists j \quad t_{ij} \in IU \\ 0, & \forall j \quad t_{ij} \notin IU \end{cases} \quad (3.3)$$

In this weighting scheme, if term t_i occurs at any position of IU , the weight of t_i is 1, otherwise its weight is 0. We can think of simple constant weighting as to judge whether a term occur in an IU, if we assume value 1 is equal to TRUE and 0 to FALSE. Then, we can use Boolean operation to derive function $f()$ in formula 3.2 and use weighting in formula 3.3 to substitute $weight()$. We also use 1 to represent TRUE and 0 to represent FALSE when we obtain the result of the formula. So we derive the following two similarity measures **Ortf** and **ANDtf**:

Ortf:

$$sim(IU, S) = OR_{t_i \in S} cw(t_i)$$

ANDtf:

$$sim(IU, S) = AND_{t_i \in S} cw(t_i)$$

When we use summation to specialize $f()$ instead of the Boolean operation

AND and OR, based on simple constant weighting in formula 3.3, we have another similarity measure distinct term frequency (**DTF**), which counts the distinct number of terms that belong to S occur in IU :

DTF:

$$sim(IU, S) = \sum_{t_i \in S} cw(t_i)$$

Apparently the similarity measure **Ortf** gives a non-zero score when any elements of S occur in IU . On the contrary, **ANDtf** gives a non-zero score when all elements of S occur in IU . **DTF**, which is more strict than **Ortf** but less strict than **ANDtf**, counts the number of distinct terms occurring both in IU and S . For example, let $S=\{a,b\}$, $IU=[a,c,b,a,d]$, **Ortf** similarity is 1, **ANDtf** similarity is 1 and **DTF** similarity is 2.

The relationship between similarity measure **DTF** and some traditional set similarity measures is shown by the following theorem:

Same Rank Theorem : For fixed sized Ius, given a term set T, to rank Ius with Jaccard's, Dice's, Overlap's and DTF similarity will have the same results.

Proof:

$|IU|$ and $|T|$ are constants.

Jaccard's similarity:

$$sim(IU, T) = \frac{|IU \cap T|}{|IU \cup T|} = \frac{|IU \cap T|}{|IU| + |T| - |IU \cap T|}$$

$$\frac{1}{sim(IU, T)} = \frac{|IU| + |T| - |IU \cap T|}{|IU \cap T|} = \frac{|IU| + |T|}{|IU \cap T|} - 1 \propto \frac{1}{|IU \cap T|}$$

Hence, $sim(IU, T) \propto |IU \cap T|$

Dice's similarity:

$$sim(IU, T) = \frac{|IU \cap T|}{|IU| + |T|} \propto |IU \cap T|$$

Overlap's similarity:

$$sim(IU, T) = \frac{|IU \cap T|}{\min\{|IU|, |T|\}} \propto |IU \cap T|$$

DTF similarity:

$$sim(IU, S) = \sum_{t_i \in S} cw(t_i) = \sum_{t_i \in S \wedge t_i \in IU} 1 = |T \cap IU|$$

Hence, all above four similarities depends on $|T \cap IU|$ so they yield same ranked results. Q.E.D.

Similarly, **DTF** reflects how many terms belong to T occur in IU , however, it cannot shows how many times they occur. Hence, we propose another similarity measure: **SumTF**. We use $tf(IU, t_i)$ to represent the term frequency of t_i in IU . The relationship between $tf(IU, t_i)$ and simple constant weighting can be shown as:

$$cw(t_i) = \begin{cases} 1, & tf(IU, t_i) \geq 1 \\ 0, & tf(IU, t_i) = 0 \end{cases}$$

We define similarity measure **SumTF** as the total frequency of all terms in S occurring in IU :

$$sim(IU, S) = \sum_{t_i \in S \wedge t_{ij} \in IU} 1 = \sum_{t_i \in S} tf(IU, t_i)$$

An essential difference between **DTF** and **SumTF** is that **DTF** counts the number of distinct terms of S in IU and **SumTF** counts the total number. For example, let IU be “the Fuji apple is an apple cultivar developed by growers at the” and S be {“Fuji”, “apple”}, the **DTF** similarity measure between IU and S is 2 since both terms in S occur in IU but **SumTF** is 3 since term “apple” occurs twice and “Fuji” occurs once.

The value of similarity **MinTF** is the minimum among term frequencies or all the terms that belongs in S occurring in IU , which is defined as:

$$sim(IU, S) = \min_{t_i \in S} \{tf(IU, t_i)\}$$

The **MinTF** relates with **ANDtf** in that both propose a conjunctive combination of the term frequencies of all terms that belongs to S occurring in IU . The relationship between **MinTF** and **ANDTF** can be shown by the following two expressions:

$$(1) \quad MinTF = 0 \Leftrightarrow ANDtf = 0$$

MinTF measure having zero values means that there is at least one term that belongs to S does not occur in IU , hence **ANDtf** is zero too, vice versa.

$$(2) \quad MinTF \geq ANDtf$$

When **MinTF** is not equal to zero, **ANDtf** is equal to one, the minimum of all term frequencies is at least one. So **MinTF** is always larger than or equal to **ANDtf**.

In this section, five constant term weighting schemes are proposed to compute the similarity of IU and the five weighting schemes produce five IU similarity measures: **Ortf**, **ANDtf**, **SumTF**, **DTF** and **MinTF**. All terms have equal weights in these five weighting schemes but different IU similarities have different ways to aggregate the term weights. Constant term weighting schemes are simpler compared with the following weighting schemes.

3.1.2 Features based term weighting

All above constant term weighting schemes do not distinguish different terms in computing the similarity between a set of terms and a sequence of terms. However, obviously, some terms are more important than others in determining how a term sequence IU relates with a set S . For example, the well-known inverse document frequency (idf) value of a term can be used to indicate the importance of a term. There are several ways to compute idf value for a term, we use the following one:

$$weight(t_{i,j}) = idf(t_i) = 1 - \frac{\lg df(t_i)}{\lg N}$$

where $df(t_i)$ denotes the document frequency of term t_i , viz. how many documents in a certain document collection contains term t_i . Since document frequency of a term only depends on the term and does not depend on the position of the term in IU, so we ignore the second suffix j and only write t_i in later $idf(t_i)$ and $df(t_i)$. N stands for total number of documents in the collection. We can derive formula 3.2 by using **idf** similarity measure:

$$sim(IU, S) = \sum_{t_i \in S \wedge t_{ij} \in IU} f(weight(t_{ij})) \quad (3.2)$$

idf:

$$\begin{aligned} sim(IU, S) &= \sum_{t_i \in S \wedge t_{ij} \in IU} f(weight(t_{ij})) \\ &= \sum_{t_i \in S \wedge t_{ij} \in IU} idf(t_{ij}) \\ &= \sum_{t_i \in S} tf(IU, t_i) \cdot idf(t_i) \end{aligned}$$

The above formula is similar with the tf-idf weighting scheme commonly used in vector space models in that for each term in S , we use tf-idf as its score, where idf value depends on the distribution of this term in the whole collection and tf refers to term frequency of this term in IU .

We notice that similarity measure **idf** does not distinguish the same terms that occur at the different positions in IU , for example, in IU “the Fuji apple is an apple cultivar developed by growers at the”, two occurrences of “apple” have the same idf value. Hence similarity measure **idf** is based on the assumption that difference occurrences of the same terms affect the similarity equally. In order to weight term considering its position in IU , we propose a position-based similarity measure. We introduce two assumptions first.

Center Term Assumption: The center term is the most important term in an IU in determining the relevance of the IU to a given topic.

Distance Assumption: The nearer two terms are in a document, the more likely they are related.

Center Term Assumption is easy to understand in that the context is extracted around the center term and provides related information of the center term. So the center term is the core of an IU and so it can be regarded as the most important term. *Distance Assumption* is based on the writer always expresses one idea at some position in a document and express another idea at another position. So the neared two terms are, it's more likely for them to present the same idea and they are more likely to related. Based on the above two assumptions, we can deduce the following observation:

Observation: Given an term set S, for term t that belongs to S, t is nearer to the center of IU, it is more important for the relevance of IU to S.

Position-based similarity measures are based on above corollary. Let the p -th word in a $(2w+1)$ -word IU be $t_{i,p}$, the center term of this IU is the w -th term. We define the simple function of distance as:

$$f_p(|p-w|) = \frac{1}{1 + \frac{|p-w+\alpha|}{1}}, \quad (\alpha \neq 0)$$

where $|p-w|$ is the distance in words between $t_{i,p}$ and the centre word (the w -th word) of IU, α is a smoothing constant. Obviously the result of above formula arranges between

0 and 1 and the results increases with the increase of $\frac{1}{|p-w+\alpha|}$. If we only use the

simple position factor as the term weight in formula 3.2 and use summation to substitute function f , we will have simple position based weighting (**SP**):

SP:

$$\begin{aligned}
 & sim(IU, S) \\
 &= \sum_{t_i \in S \wedge t_{ij} \in IU} f(weight(t_{ij})) \\
 &= \sum_{t_i \in S \wedge t_{ij} \in IU} (f_p(|j-w|)) \\
 &= \sum_{t_i \in S \wedge t_{ij} \in IU} \left(\frac{\frac{1}{|p-w+\alpha|}}{1 + \frac{1}{|p-w+\alpha|}} \right)
 \end{aligned}$$

We can also combine more than one weighting schemes together to derive new similarity measures. For example, if we assume that similarity between an IU and S depends on both generality of term, measured by idf value, and position, measured by **SP**. We should use conjunctive operation to combine the two as:

$$weight(t_{i,j}) = idf(t_i) \wedge f_p(|j-w|)$$

If we use multiplication to substitute the conjunction operation “ \wedge ” in above formula and use summation to derive the function $f()$ in formula 3.2, we have another similarity measure defined as:

IDF-POST:

$$\begin{aligned}
& sim(IU, S) \\
&= f_{t_i \in S \wedge t_{ij} \in IU} (weight(t_{ij})) \\
&= \sum_{t_i \in S \wedge t_{ij} \in IU} (idf(t_i) \cdot f_p(|j - w|)) \\
&= \sum_{t_i \in S \wedge t_{ij} \in IU} (idf(t_i) \cdot \frac{1}{|j - w + \alpha|})
\end{aligned}$$

In this section, three feature based weighting schemes are proposed to compute the similarity of IU and the three weighting schemes produce three IU similarity measures: **idf**, **SP**, and **IDF-POST**. Terms are weighted according to their idf value and position (distance to the middle of IU) respectively in **idf** and **SP**. **IDF-POST** similarity measure is a combined term weighting scheme based on idf value and position of terms. IU similarity is the aggregation of weights of the terms and all three similarity measures use summation operation as the aggregation. Feature based term weighting schemes are more complex compared with constant weighting schemes.

3.1.3 Fuzzy set based similarity

In subsection 3.1.1 and 3.1.2, we have made use of summation and Boolean operations to substitute the aggregating function $f(\cdot)$ in formula 3.2. The choice of aggregating function relies on the dependency of the final similarity between IU and term set S on occurrence of each term in S and in IU . More generally, we can use a fuzzy set operation f_{fuzzy} to substitute the function f in formula 3.2. Then the similarity between IU and a term set is:

$$sim(IU, S) = f_{fuzzy}_{t_i \in S \wedge t_{ij} \in IU} (weight(t_{ij})) \quad (3.3)$$

where f_{fuzzy} is a fuzzy operation, including fuzzy intersection and fuzzy union.

Let $i_w(a,b)$ is a fuzzy operation of combing two real number a and b (w is a parameter of this operation), $sim(IU, S)_j$ is the similarity of IU and S after considering $t_{i,j}$, term t_i occurring at the j -th position in IU , $\bar{w}()$ is a normalized (transferred into the interval $[0,1]$ to meet the requirement of fuzzy operation) weight of term $t_{i,(j+1)}$, then we give a iterative definition of similarity between IU and S :

Table 0.□.□ For the term at the first position of IU , the similarity is defined by:

$$sim(IU, S)_1 = \bar{w}(t_{p1})$$

ii. For the rest terms of IU :

$$sim(IU, S)_{j+1} = i_w(sim(IU, S)_j, \bar{w}(t_{i(j+1)})) \quad (j > 1)$$

For example, if we use [Dombi 82] fuzzy set conjunction (AND or intersection), which is defined by:

$$i_p(a,b) = \frac{1}{1 + \left[\left(\frac{1}{a} - 1 \right)^p + \left(\frac{1}{b} - 1 \right)^p \right]^{\frac{1}{p}}}$$

where values of the parameter p lie in the open interval $(0, +\infty)$. The (ii) of above iterative definition is derived into:

$$\begin{aligned} & sim(IU, S)_{j+1} \\ &= i_p(sim(IU, S)_j, \bar{w}(t_{i(j+1)})) \\ &= \frac{1}{1 + \left[\left(\frac{1}{sim(IU, S)_j} - 1 \right)^p + \left(\frac{1}{\bar{w}(t_{i(j+1)})} - 1 \right)^p \right]^{\frac{1}{p}}} \end{aligned}$$

We can also use extended Boolean conjunction to specialize the fuzzy set function $f_{fuzzy}(\cdot)$ in formula 3.3. Similar with the decision parameter p of Dombi intersection operator, extended Boolean conjunction also has a decision parameter p which can be tuned to achieve the optimal decision hard/soft level. Based on extended Boolean conjunction, formula 3.3 is derives into:

$$sim(IU, S) = 1 - \sqrt[p]{\frac{1}{SUMtf} \sum_{i,j} (1 - \bar{w}(t_{ij}))^p}$$

We noticed that [Wu 07] tried to discover the best aggregation operator to combine the context scores in a document. They found that when $p=1$, the result of extended Boolean AND is the best among other setting of p ($p=5, 10, 20, 40, \infty$). When $p=1$, above formula is degenerated into the arithmetic average of the weights of all the terms:

$$sim(IU, S) = \frac{1}{SUMtf} \sum_{i,j} \bar{w}(t_{ij})$$

This formula is suitable for term feature-based weighting schemes. Because for constant weighting scheme, the weights of different terms are same and the arithmetic average of the summation of the weights is just the weight of each term, which cannot discriminate different Ius.

In this section, two fuzzy set weighting schemes are proposed to compute the similarity of IU and the two weighting schemes produce two IU similarity measures by deriving the aggregation function into Dombi fuzzy set conjunction (intersection) operator and extended Boolean conjunction operations.

3.2 Graph-based Model

Fuzzy model is dealing with the computation of similarity between a term sequence IU and a term set. It considers each single term that occurs both in IU and in term set as the basis element. Graph-based model is dealing with the computation of the similarity between a term sequence IU and one or two term sets by considering a pair of terms as the basic element in calculation. This pair of terms is formulated into an edge linking two vertices. Furthermore, graph-based model is able to deal with the similarity between a term sequence IU and two terms sets: one set contains topic entity terms and the other set reflects the information other than topic entity, for example, discourse type, which will be completely introduced in Chapter 4.

The Graph-based Model is based on the assumption that the relevance of IU depends on the relationship among terms in IU and the relationship can be represented by each two of the terms. The different types of relationships are represented by a graph $G(V,E)$

3.2.1 IU Graph

The terms in an IU are classified into two groups, one is directly related with topic entity and the other is related with other desirable information that makes an IU relevant. We use the terms related with discourse type as an example to formulate IU graph and graph-based model. In fact, the set of terms other than the topic entity can be

very general.

We assume that we previously know the discourse type of a given topic and this discourse type is presented by a set of single terms, namely discourse terms, in documents. The introduction of how to obtain and evaluate these discourse terms is in Chapter 4. In this chapter, we assume that a set of discourse terms has been already generated for us to use. We know that some discourse types are presented by some phrases even sentences rather than single terms. In order to simplify our model, we assume the discourse terms are all single terms.

Given a topic, let T be a term set containing topic entity terms which are always selected from title section of the TREC topics. Let D be a term set containing discourse terms of the discourse type of this topic. We assume that all terms in T are distinct, all terms in D are distinct, T and D are disjointed. We propose our graph-based model (GM) based on the assumption that the similarity between an IU and a topic depends on some selected pairs of terms in the IU and these pairs of terms belong to T or D .

We use a graph to represent these term pairs. An IU graph is defined as a graph $G(V, E)$ where V is the set of vertices and E is the set of edges. Each vertex in V is a term at a particular position in the IU and $V \subseteq T \cup D$. Each edge in E is an edge linking two vertices of V . The similarity of an IU between and a topic, written by a query q , is:

$$sim(IU, q) = f(G(V, E)) \quad (3.4)$$

where function $f()$ defines how we compute the similarity between IU and q based on IU graph $G(V, E)$.

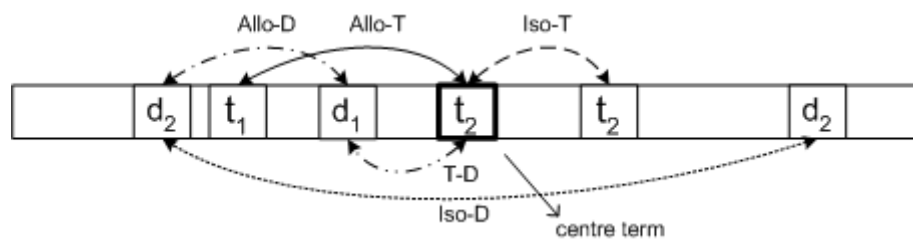
We defined six types of edges according to the different types of terms that an edge

links, as shown in Table 3.1. We use $t_{i,p}$ to denote the i -th term in T that occurs at the p -th position in IU and $d_{j,q}$ to denote the j -th term in D that occurs at the q -th position in IU.

Table 3.1 Edges denotation and description

Edge Name	Denotation	Description
iso-T edge	$\text{edge}(t_{i,p}, t_{i,q})$	between the same topic terms
allo-T edge	$\text{edge}(t_{i,p}, t_{j,q})$	between the different topic terms
T-D edge	$\text{edge}(t_{i,p}, d_{j,q})$	between a topic term and a discourse term
iso-D edge	$\text{edge}(d_{i,p}, d_{j,q})$	between the same discourse terms
allo-D edge	$\text{edge}(d_{i,p}, d_{j,q})$	between the different discourse terms
self-cycle	$\text{edge}(t_{i,p}, t_{i,p})$	between a term and itself

Figure 3.1 Five types of edges



In figure 3.1, we present an IU and mark the first five edges with different types of lines. In order to make the figure simple, we ignore the self-cycle edges around each term. For each term, we only write the first suffix number to distinguish different terms

and we ignore the second suffix number for the sake of brevity. This IU contains two distinct topic entity terms t_1 (occurs once) and t_2 (occurs twice) and two distinct discourse terms d_1 (occurs once) and d_2 (occurs twice). The centre term of this IU is the first occurrence of term t_2 .

Given an IU IU , a topic entity term set T and a discourse term set D of a given topic, we will compute the similarity between IU and topic based on the different types of edges in the IU graph of IU . We use $iso-T(E)$ to represent the set of iso-T edges, a subset of E . In the similar way we represent other five subsets of E that contains the rest five types of edges. We use $\varepsilon(E')$ to represent the relevant evidence score of an edge set E' . Hence, $\varepsilon(iso-T(E))$ denotes the relevant evidence score of iso-T edges of IU graph $G(V,E)$. Then we use an aggregation function agg_2 to combine the scores from all types of the edges as follows:

$$f(G(V, E)) = agg_2 \left\{ \begin{array}{l} \varepsilon(iso-T(E)), \varepsilon(allo-T(E)), \varepsilon(TD(E)), \\ \varepsilon(iso-D(E)), \varepsilon(allo-D(E)), \varepsilon(cycle(E)) \end{array} \right\} \quad (3.5)$$

Formula 3.5 is general to aggregate relevant evidence score of all possible types of edges in G . We can also choose some certain types of edges for aggregation, which compose a subgraph of G . We will introduce the subgraphs of $G(V,E)$ and the computation of the similarity between an IU and a topic can depends on a subgraph of the IU graph.

In this subsection, we introduce the graphic representation of an IU: IU graph and the different types of edges that compose an IU graph. There are six types of edges, which are classified by the vertices that are linked. The similarity of an IU can be computed based on the IU graph of this IU. In details, the similarity of an IU graph can

be derived into the relevant evidence score of each edge or the selected edges (maybe more important than others in representing the relevant information) of the IU graph. When not all the edges are counted for computing the similarity score of an IU graph, the concept of subgraph of an IU graph appears. Subsection 3.2.2 will introduce subgraph of an IU graph.

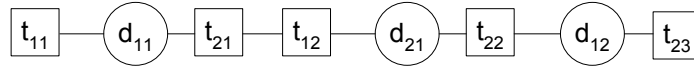
3.2.2. Subgraph of an IU graph

A subgraph $G'(V',E')$ of an IU graph $G(V,E)$ is composed of some certain types of edges of E so $V' \subseteq V$ and $E' \subseteq E$.

We use an IU, given in figure 3.1, as an example to show the possible subgraphs of an IU graph. For simplification, we change a given IU into a term sequence that only composes the terms that belongs to T or D . It is generated by removing all terms that do not belongs to the union of T and D and we keep the original order of the terms as they occur in the former IU. In order to make the figures easy to see, we write a topic entity term as t_{ij} instead of $tt_{i,j}$ and write a discourse type term d_{ij} instead of $dt_{i,j}$ in the figures. The topic entity terms are in the squares and discourse type terms are in the circles. In order to make the following figures simple, in Figure 3.2 and all subgraph figures in Table 3.2, the second suffix number, say j , of each term (t_{ij} or d_{ij}) only stands for different occurrence of term t_i or d_i rather than the original meaning --- term position of IU. Then, if we write t_{21} , it means the first occurrence of term t_2 but in real case, when the other terms have not been removed, we should write it as $t_{2,p}$, where p is its actual position in the IU. In the IU shown in figure

3.1, topic entity term t_1 occurs twice, t_2 occurs for three times. Discourse term d_1 occurs twice and d_2 occurs once.

Figure 3.2 Example of Representation of an IU



For the sake of brevity, we ignore the self-cycle edges around all terms. In the following sections, we also ignore iso-D edges and allo-D edges because first there are fewer discourse terms than topic entity terms. Second, we do not want to deal with the relationship between two identical (iso-D edges) or two different (allo-D) discourse terms in graph-based models. We will formulate the relationship between discourse terms as features in next chapter. Hence, based on whether a subgraph contains iso-T edge, allo-T edge and T-D edge, there are eight types of subgraph, which are shown in Table 3.2. The first three columns of Table 3.2 respectively indicate whether a subgraph contains iso-T edge, allo-T edge and T-D edge. The fifth column contains some description of the corresponding subgraph and a figure. In the figure, we use a block arrow to denote all edges in a complete bipartite graph, such as in TD graph, the block arrow denotes all possible T-D edges linking between each t_{ij} vertex and each d_{ij} vertex. We use broken line to denote iso-T edges.

Table 3.2 Subgraphs of an IU graph (part I)

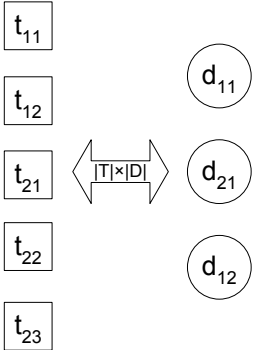
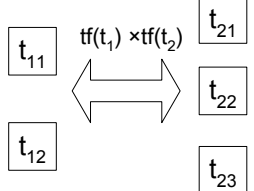
iso-T edge	allo-T edge	T-D edge	Subgraph Name	Description of the graph and figure
no	no	no	Scattered vertices	All terms are independent: traditional vector space model, fuzzy model.
No	no	yes	TD graph	<p>A bipartite graph between T and D</p> 
no	yes	no	allo-T graph	<p>A bipartite graph between T and T</p> 

Table 3.2 Subgraphs of an IU graph (part II)

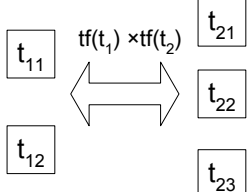
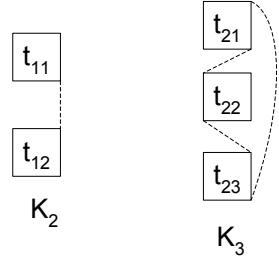
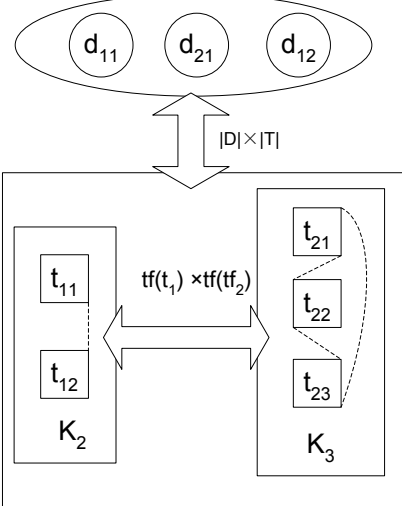
iso-T edge	allo-T edge	T-D edge	Subgraph Name	Description of the graph and figure
no	yes	yes	allo-T graph with TD	
yes	no	no	iso-T graph	<p>m independent complete graphs</p> <p>$K_{tf(IU,t)}$</p> <p>(note: m is the number of distinct terms)</p> 

Table 3.2 Subgraphs of an IU graph (part III)

iso-T edge	allo-T edge	T-D edge	Subgraph Name	Description of the graph and figure
yes	no	yes	iso-T graph with TD	
yes	yes	no	Pan-T graph	
yes	yes	yes	Pan-graph	 <p>The diagram illustrates a Pan-graph structure. At the top, an oval contains three nodes labeled d_{11}, d_{21}, and d_{12}. Below this oval is a double-headed arrow labeled $D \times T$. The main part of the diagram is enclosed in a large rectangle and contains two subgraphs, K_2 and K_3. Subgraph K_2 on the left contains nodes t_{11} and t_{12}. Subgraph K_3 on the right contains nodes t_{21}, t_{22}, and t_{23}. A double-headed arrow labeled $tf(t_1) \times tf(t_2)$ connects the two subgraphs. Dotted lines indicate internal connections within K_3 between t_{21} and t_{22}, and between t_{22} and t_{23}.</p>

There are eight types of subgraphs of an IU graph. A group of scattered vertices is the simplest one. There is no edge formed in this subgraph, which assume that the terms are mutually independent. We can think of traditional vector space model, fuzzy models introduced in section 3.1 etc. into this subgraph, In fact, it's not a graph and we introduce it just to keep a complete and consistent formulation for subgraphs of IU graph. As for T-D graphs, allo-T graphs and iso-T graphs, each of them contain one type of edges corresponding to their graph names. Allo-T graph with TD and iso-T graph with TD are built after we add T-D edges to allo-T graph and iso-T graph. Pan-T graph contains allo-T edges and iso-T edges. Pan-graph contains all three types of edges.

The similarity between an IU and a given topic can be determined by a certain subgraphs. We take allo-T graph as an example to derive formula 3.4 and 3.5:

$$sim(IU, q) = f(G'(V', allo - T(E))) = \varepsilon(allo - T(E)) \quad (3.6)$$

In this subsection, we propose the concept of IU subgraph and theoretically investigate the possible types of subgraphs that can be formed from a complete IU graph. However, not all edges belonging to the same type have to be considered when we are calculating IU similarity. We need to select the edges according to other criterions besides edge type. In next subsection, we will introduce different edge selection schemes.

3.2.3 Edges Selection Schemes

The formula 3.6 is one way to compute the similarity between IU and a topic, which is based on a subgraph, namely allo-T graph. This subgraph may contain many edges and these edges have different significance in computing the relevant evidence score of the subgraph. So the problem of edges selection appears. The selection of edges is based on *Center Term Assumption* and *Distance Assumption* introduced in section 3.1.2.

An edge links two vertices and each vertex represents one term I. So sometimes we say an edge links two terms for the sake of brevity. According to above two assumptions, the selection of edges considers two factors. One is whether the two terms linked by an edge contain the center term of IU. The other factor is whether the two terms linked by an edge are the closest two terms. We introduce the edges selection schemes for different types of edges as follows.

Allo-T Edge

Let an IU contain $2w+1$ terms, we define the following five edge selection schemes for allo-T graph and each of them select a set of edges:

(1) *All(at01)*: All allo-T edges are selected:

$$\{edge(t_{i,p}, t_{j,q})\}$$

(2) *Center (at02)*: Select the edges that link with the center term of IU (the center term is at the w -th position of IU. If there is no allo-T edge linking with the center term, empty set is obtained.):

$$\{edge(t_{i,p}, t_{j,q}) : w \in \{p, q\}\}$$

(3) *Nearest for each pair (at03)*: Among each possible pair of two different topic entity

terms, keep the edge linking the nearest pair:

$$\{edge(t_{i,p}, t_{j,q}) : (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(4) *Nearest to centre (at04)*: among edges selected by *center*, select the edge linking the nearest pair:

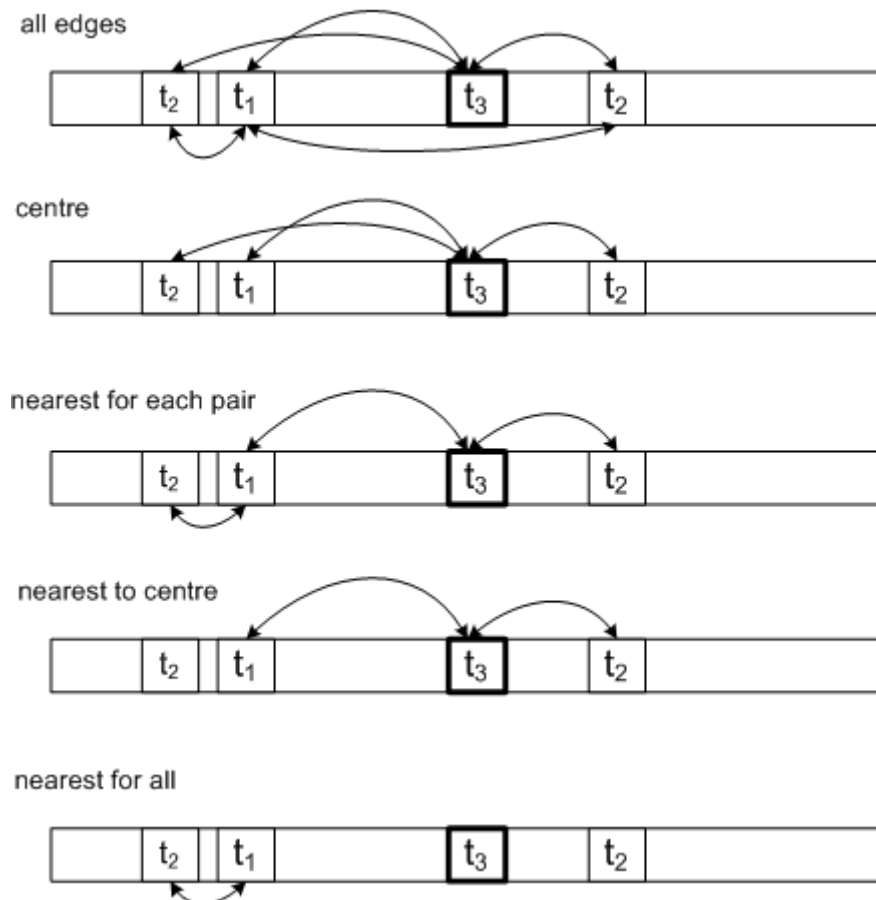
$$\{edge(t_{i,p}, t_{j,q}) : w \in \{p, q\}, (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(5) *Nearest for all (at05)*: select the edge linking the nearest pair:

$$\{edge(t_{i,p}, t_{j,q}) : (i, j, p, q) = \arg \min_{(i,j,p,q)} |p - q|\}$$

Let's take an IU consisting of term t_1 , t_2 and t_3 as an example. The centre term of the IU is t_3 . Figure 3.2 shows the five edge selection schemes.

Figure 3.2 Example of Edge Selection Schemes for allo-T graph



Iso-T Edge

We can define similar five edge selection schemes for iso-T edges.

(1) *All(it01)*: All iso-T edges are selected:

$$\{edge(t_{i,p}, t_{i,q})\}$$

(2) *Center (it02)*: Select the edges that link with the center term of IU (the center term is at the w -th position of IU. If there is no iso-T edge linking with the center term, empty set is obtained.):

$$\{edge(t_{i,p}, t_{i,q}) : w \in \{p, q\}\}$$

(3) *Nearest for each pair (it03)*: Among each possible pair of two same topic entity terms, keep the edge linking the nearest pair:

$$\{edge(t_{i,p}, t_{i,q}) : (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(4) *Nearest to centre (it04)*: among edges selected by *center*, select the edge linking the nearest pair:

$$\{edge(t_{i,p}, t_{i,q}) : w \in \{p, q\}, (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(5) *Nearest for all (it05)*: select the edge linking the nearest pair:

$$\{edge(t_{i,p}, t_{i,q}) : (i, p, q) = \arg \min_{(i,p,q)} |p - q|\}$$

T-D Edge

Edge selection scheme for T-D edges is a little different from allo-T and iso-T edges since the center term of an IU is always a topic entity term. So when the selection scheme is concerned about the center of IU, the topic entity term linked by a edge is the center term. Edge selection scheme for T-D edges are defined as:

(1) *All(td01)*: All iso-T edges are selected:

$$\{edge(t_{i,p}, d_{j,q})\}$$

(2) *Center (td02)*: Select the edges that link with the center term of IU (the center term is at the w -th position of IU. Remember the center of IU is always a topic entity term.):

$$\{edge(t_{i,p}, d_{j,q}) : w = p\}$$

(3) *Nearest for each pair (td03)*: Among each possible pair of a topic entity terms and a discourse type term, keep the edge linking the nearest pair:

$$\{edge(t_{i,p}, d_{j,q}) : (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(4) *Nearest to centre (td04)*: among edges selected by *center*, select the edge linking the nearest pair:

$$\{edge(t_{i,p}, d_{j,q}) : w \in \{p, q\}, (p, q) = \arg \min_{(p,q)} |p - q|\}$$

(5) *Nearest for all (td05)*: select the edge linking the nearest pair:

$$\{edge(t_{i,p}, d_{j,q}) : (i, j, p, q) = \arg \min_{(i,j,p,q)} |p - q|\}$$

We give example on how to apply these schemes on a subgraph. We can apply different scheme on different edges. We take the subgraphs described in Table 3.2 as example, we apply *nearest for all* scheme for the three types of edges, because it select only one edge. Actually, it is equal to **at05**, **it05** and **td05** on each of the subgraphs.

In this subsection, we introduce five edge selection schemes for each type of edges. The selection considers different factors, including whether an edge links with the middle term of an IU, or whether an edge is the shortest one (or linking the nearest two terms) among all possible term pairs, etc. The strategies behind the five edge selection schemes are quite different, which are evaluated by the following experiments. Obviously, different edge selection schemes produce different number of edges.

3.2.4 Relevant Evidence Score of an Edge

From the definition of IU graph, we know that an edge links two vertices in an IU graph or in the subgraphs of an IU graph. The two linked vertices represent two occurrences of term(s). The relevant evidence score of an edge quantifies the evidence indicating the presence of relevant information by the two occurrences of term(s). Different types of edges function differently in determining the content of an IU that the IU graph represents. We will respectively discuss the computation of relevant evidence score for each type of edges.

Allo-T Edge

An allo-T edge $edge(t_{i,p}, t_{j,q})$ links two different topic entity terms. The two terms can mutually provide information to disambiguate and present a more definite sense together. Aside from specificity, which indicates the independent feature of each term alone, we also consider the order of two terms and the distance between two terms. Under some circumstances, the order of two terms greatly determines the sense presented by the two terms greatly, such as “social” and “security”. Sometime, such determination is not obvious, such as “term limits” and “term limits”. The distance of two terms occurring in IU determines the association of two terms, which can reflect how one term is likely to provide information to the other term. Overall, the relevant evidence score of an allo-T edge is conditioned by the following factors:

- *Specificity*: the more specific a term is, the less likely it is polysemous. So it is more likely that the meaning of the term in the document is the same as that in the query. Specificity factor is measured by function $IDF()$;
- *Order*: if these terms are in the same order as they are in the query then it is more likely that the matched terms have the same meaning as the matched query terms. This is denoted by function $order()$;
- *Distance*: the matched terms are nearer to each other, then it is more likely that the matched terms have the same meaning as the matched query terms. This is also denoted by function $f_p()$;

Then, based on the above three factors, the relevance evidence score of an allo-T edge that links vertex $t_{i,p}$ and $t_{j,q}$ is:

$$\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = IDF(t_i, t_j) \wedge \text{order}(t_{i,p}, t_{j,q}) \wedge f_p(|p - q|)$$

The above formula computes the relevance evidence score of an allo-T edge in IU graph, in which symbol “ \wedge ” denotes a conjunctive function because the relevance evidence score conjunctively depends on the three factors. We derive this conjunctive function by multiplication in later experiments. Factor $IDF()$ is derived into:

$$IDF(t_i, t_j) = idf(t_i) \wedge idf(t_j)$$

where “ \wedge ” also stands for a conjunctive function, which can be specialized by multiplication or fuzzy AND operation.

Factor $order()$ will be determined based on the assumption that the order of two terms is able to affect the meaning of the phrases that inclusively or exclusively contain these two terms. Let assume that when t_i occurs before t_j , it is better to present

the topic entity. We derive *order* factor based on the second suffix of each term as:

$$order(t_{i,p}, t_{j,q}) = \begin{cases} 1, & p < q \\ \omega, & p > q \end{cases} \quad (0 < \omega < 1)$$

where ω is a positive constant that is smaller than one.

We can also use the simple function of distance that we used in section 3.1.2 to derive *distance* factor:

$$f_p(|p - q|) = \frac{1}{|p - q + \alpha|}, \quad (\alpha \neq 0)$$

Based on above derivation of the three factors, we combine all three to compute the relevant evidence score for an allo-T edge. Formula 3.6 computes the similarity between an IU and a topic based on the allo-T subgraph of the IU graph. So this similarity is based on all relevant evidence scores of all allo-T edges, which can be further derived into formula 3.7:

$$\begin{aligned} & sim(IU, q) \\ &= f(G'(V', allo - T(E))) \\ &= \varepsilon(allo - T(E)) \\ &= agg_3[ESS[\varepsilon(edge(t_{i,p}, t_{j,q}))]] \\ &= agg_3[ESS[IDF(t_i, t_j) \wedge order(t_{i,p}, t_{j,q}) \wedge f_p(|p - q|)]] \quad (3.7) \\ &(t_i, t_j \in T, \quad t_{i,p}, t_{j,q} \in IU, \quad edge(t_{i,p}, t_{j,q}) \in ESS) \end{aligned}$$

In formula 3.7 a certain edge selection scheme (any of **at01**, **at02**, ..., **at05**) is applied on selecting some allo-T edges to build a edge set *ESS* and scores of these selected edges are aggregated by a function *agg₃*.

Iso-T Edge

An iso-T edge $edge(t_{i,p}, t_{i,q})$ links two occurrences, at the p -th position and q -th position, of a topic entity term t_i . Just as the specificity factor used for allo-T edge, the

specificity of the terms linked by iso-T edge is also a factor to measure the relevance evidence score. The distance between the two term occurrences reflects the extension of the meaning that the term entity term expresses. If there is a different topic entity term t_j occurring between the two occurrences of a topic entity term t_i , it's very likely that the text beginning from the first occurrence of t_i to the second occurrence of t_i is relevant to term t_i and term t_j . We use a factor *interlace* to measure how three occurrences mentioned above exist in an IU. In conclusion, the relevant evidence score of an iso-T edge is conditioned by the following factors:

- *Specificity*: Since iso-T edge links two occurrence of the same term, the specificity factor of an iso-T edge depends on only one term, which is measured by function $IDF()$;
- *Distance*: the distance factor of an iso-T edge is to measure the extension of one topic entity term in an IU. So we will derive it differently with the distance factor of the allo-T edge. It is measured by the function $f_{pe}()$;
- *Interlace*: this factor depends on how many distinct topic entity terms occur between two occurrences of a topic entity terms. This factor is measured by $itl()$.

Then, based on the above three factors, the relevance evidence score of an iso-T edge that links vertex $t_{i,p}$ and $t_{i,q}$ is:

$$\varepsilon(\text{edge}(t_{i,p}, t_{i,q})) = IDF(t_i) \wedge f_{pe}(|p - q|) \wedge itl(t_{i,p}, t_{i,q})$$

In above formula symbol “ \wedge ” denotes a conjunctive function because the relevance

evidence score conjunctively depends on the three factors. We specialize this conjunctive function using multiplication in later experiments. Factor $IDF()$ is derived into:

$$IDF(t_i) = idf(t_i)$$

where $idf()$ is the idf value of term t_i .

Factor distance depends on the distance of two occurrences of a topic entity term and when the distance is large, the extension of the topic entity term is larger. Hence,

$$f_{pe}(|p - q|) = \alpha \cdot |p - q|, \quad (\alpha > 0)$$

$itl(t_{i,p}, t_{i,q})$ is equal to the number of distinct topic entity terms occurring between the p -th position and q -th position in the IU.

Based on above derivation of the three factors, we combine all three to compute the relevant evidence score for an allo-T edge. We derive the following formula to compute the similarity between an IU and a topic based on the iso-T subgraph of the IU graph. So this similarity is based on relevant evidence scores of all selected iso-T edges in set ESS generated by a certain edge selection scheme.

$$\begin{aligned} & sim(IU, q) \\ &= f(G'(V', iso - T(E))) \\ &= \varepsilon(iso - T(E)) \\ &= \underset{ESS}{agg}_3[\varepsilon(edge(t_{i,p}, t_{i,q}))] \\ &= \underset{ESS}{agg}_3[IDF(t_i) \wedge f_{pe}(|p - q|) \wedge itl(t_{i,p}, t_{i,q})] \\ &(t_i \in T, \quad t_{i,p}, t_{i,q} \in IU, \quad edge(t_{i,p}, t_{i,q}) \in ESS) \end{aligned}$$

T-D Edge

An T-D edge $edge(t_{i,p}, d_{i,q})$ links one topic entity term $t_{i,p}$ and one discourse

type term $d_{i,q}$. So this edge contains two types of information: information concerned with the topic entity and information concerned with the discourse type. Just as the specificity factor used for allo-T edge and iso-T edge, we also use specificity factor to measure the topic entity term in the T-D edge. We use factor *confidence* to measure the probability that the IU has the same discourse type as the topic with the presence of discourse type term $d_{i,q}$. The distance between $t_{i,p}$ and $d_{i,q}$ also reflects the association of two terms. Hence, the relevant evidence score of a T-D edge is conditioned by the following factors:

- *Specificity*: as what discussed above we need to know how specific the topic terms is, measured by function $IDF()$.
- *Confidence*: discourse type terms are quite different in the ability to indicate a discourse type. For example, “because” is more reliable than “since” and “as” to judge whether a sentence states reason. We use this factor to measure how confident we believe the IU has the same discourse type as the topic requires with the presence of this discourse term. It is measured by the function $conf()$.
- *Distance*: if the discourse type term $d_{i,q}$ is near to the topic entity term $t_{i,p}$, it's very likely that the discourse type presented by $d_{i,q}$ is related with the topic entity term. Nearer they are, more likely they relate each other. It is measured by the function $f_p()$.

Therefore, based on the above three factors, the relevance evidence score of an

T-D edge that links vertex $t_{i,p}$ and $d_{j,q}$ is:

$$\varepsilon(\text{edge}(t_{i,p}, d_{j,q})) = IDF(t_i) \wedge \text{conf}(d_{j,q}) \wedge f_p(|p - q|)$$

where $IDF()$ denotes the idf value of topic entity term $t_{i,p}$ and $f_p()$ is the simple function of distance we used for computing the relevance evidence score of allo-T edge.

Similarly, we derive the following formula to compute the similarity between an IU and a topic based on the T-D subgraph of the IU graph. So this similarity is based on relevant evidence scores of all selected T-D edges in set ESS generated by a certain edge selection scheme.

$$\begin{aligned} & \text{sim}(IU, q) \\ &= f(G'(V', T - D(E))) \\ &= \varepsilon(T - D(E)) \\ &= \underset{ESS}{\text{agg}_3}[\varepsilon(\text{edge}(t_{i,p}, d_{j,q}))] \\ &= \underset{ESS}{\text{agg}_3}[IDF(t_i) \wedge \text{conf}(d_{j,q}) \wedge f_p(|p - q|)] \\ & (t_i \in T, d_j \in D, t_{i,p}, d_{j,q} \in IU, \text{edge}(t_{i,p}, d_{j,q}) \in ESS) \end{aligned}$$

In this subsection, we introduce the calculation of relevant evidence score for different types of edges. The score calculation reflects and considers the different characters of different types of edges. As the basic elements, these scores are later aggregated together to form a score for IU similarity based on graph.

3.3 Experimental Results

3.3.1 Experiment Setup

We use the TREC text research collections in all the experiments mentioned in this thesis. The TREC text research collections include materials from the Financial Times Limited (1992-1994, approximate 210,000 documents and 565MB), the Federal Register (1994, approximate 55,000 documents and 395MB), the Foreign Broadcast Information Service text (1996, 130,000 documents and 470MB) and the Los Angeles Times (1989-1990, 130,000 documents and 475MB). In total, TREC collection includes about 525,000 documents and 1,905 megabytes of text.

The Federal Register, abbreviated FR, is the official journal of the federal government of the United States that contains most routine publications and public notices of government agencies, which includes new/final rules and regulations, and notices of meetings and adjudicatory proceedings. The Foreign Broadcast Information Service text is produced by monitoring, collecting and translating within the U.S. government openly available news and information from media sources outside the United States.

The Financial Times text contains two categories: one covers domestic and international news, editorial commentary on politics and economics from their journalists and opinion pieces from globally renowned leaders, policymakers, academics and commentators, the other consists of financial data and news about companies and markets. The Los Angeles Times is a daily newspaper published in Los Angeles, California and it is the second-largest metropolitan newspaper in circulation

and the fourth most widely distributed newspaper in the United States. Compared with the Financial Times text, the Los Angeles Times contains a wider variety of material, including editorial opinions, criticism, persuasion and op-eds, reviews of radio, movies, television, plays and restaurants etc.

Therefore, it is well acknowledged that the document set used in TREC has the diversity of subject matter (called topic in TREC), word choice, literary styles, document formats, etc. in order to make the retrieval results to be representative of the performance in the real task.

Our experiments are based on TREC queries. By studying the 250 topics of TREC Robust Track 2004 (including topics of TREC-6, TREC-7, TREC-8 and 100 new queries, all in English), we found that some queries have the same type of information need, though the topic entities are quite different. For example, topic 605 requires “What are the *pros and cons* of Great Britain’s universal health care system?” and topic 624 requires “What are the *pros and cons* of developing the Strategic Defense Initiative (SDI) also known as “Star Wars”?” Apparently, these two queries need the same feature or property of the different entities. It’s this discovery that encourages us to do the research on so-called “discourse type” of the information need. Chapter 4 includes very detailed introduction of the definition, recognition and evaluation on different discourse types. Hence in this chapter we don’t explain too much on discourse types. In this section, we report the experiments on the IU similarity models based on the queries grouped by discourse type. In Chapter 4, we will introduce how to

manually group the queries and how to justify that the manually grouping is reliable and consistent. Here, we only briefly introduce what the discourse types are and which queries are grouped in each of the discourse types.

In order to provide a general idea of the level of the performance of our baseline retrieval, we offer the results of some participants of TREC 2004 Robust Retrieval Track of the 249 queries [Voorhees 04]. Since only top ten results are provided, we select the groups that rank the 1st, the 5th and 10th as reference. And the retrieval performance in MAP of the 1st, the 5th and 10th groups are 0.333, 0.282 and 0.231 and ours is 0.296. The retrieval performance in “precision at ten documents (P@10)” of the 1st, the 5th and 10th groups are 0.513, 0.437 and 0.414 and ours is 0.457. We find that if ranked by MAP, the performance of our baseline lies between the 4th and 5th group.

Advantage/disadvantage

There are eight queries that require information on the advantages and/or disadvantages of a certain technology, proposal or some policies. In the “Description” parts of these eight queries, it’s clearly written that the information need is the “advantages and/or disadvantages” or “pros and cons” of the corresponding topic entity. Table 3.3 lists these eight queries and their retrieval performance in MAP by our baseline retrieval engine: the BM25 term weight of the 2-Poisson model [Robertson 94] using the standard parameter setting [Robertson 97] (i.e., $k_1 = 1.2$ and $b = 0.75$) with passage-based retrieval and pseudo relevance feedback (PRF). In experiments reported in the following sections, the retrieved list is generated by this baseline retrieval engine and re-ranked by combining with the measures obtained from our IU similarity

models.

Table 3.3 Topics with discourse type *advantage/disadvantage* and their retrieval performance

Discourse Type: advantage/disadvantage	
Topic ID and Title of Query	MAP
308 Implant Dentistry	0.124
605 Great Britain health care	0.176
608 taxing social security	0.133
624 SDI Star Wars	0.299
637 human growth hormone (HGH)	0.396
654 same-sex schools	0.041
690 college education advantage	0.004
699 term limits	0.496
Mean MAP	0.209

Table 3.4 Topics with discourse type *country* and their retrieval performance

Discourse Type: country	
Topic ID and Title of Query	MAP
318 Best Retirement Country	0.003
428 declining birth rates	0.356
438 tourism, increase	0.320
445 women clergy	0.309
632 southeast Asia tin mining	0.120
689 family-planning aid	0.063
Mean MAP	0.195

Country

There are six queries that require information on the names of countries that can satisfy some certain conditions or have some special properties. For example, the “Description” part of topic number 428 is “*Do any countries other than the U.S. and China have a declining birth rate?*” and that of topic number 438 is “*What countries are experiencing an increase in tourism?*”. The discourse type “advantages and/or

disadvantages” and “country” are both attributes of an entity. We also know that a name of country is always a proper name with capital letters. These six queries and their retrieval performance in MAP are listed in Table 3.4.

Reason

According to the *Random House Unabridged Dictionary* published in 2006, causality is defined as a necessary relationship between one event, which is (called cause) and another event (called effect) which is the direct consequence (result) of the first. There are nine queries that require information on the reasons or causes of some certain events or phenomenon. The Description parts of these queries begin with “*Determine the reasons why...*”, “*What are the causes of...*” or “*Find documents that discuss reasons why...*”. These nine queries and their retrieval performance in MAP are listed in Table 3.5.

Table 3.5 Topics with discourse type *reason* and their retrieval performance

Discourse Type: reason	
Topic ID and Title of Query	MAP
333 Antibiotics Bacteria Disease	0.386
397 automobile recalls	0.481
436 railway accidents	0.156
628 U.S. invasion of Panama	0.224
636 jury duty exemptions	0.185
639 consumer on-line shopping	0.242
669 Islamic Revolution	0.052
670 U.S. elections apathy	0.187
673 Soviet withdrawal Afghanistan	0.104
Mean MAP	0.224

Impact

There are twelve queries that require information on the impact of some certain events or phenomenon. The Description parts of these queries begin with “*What impact...*”, “*Find information on ...’s impact on ...*” or “*Find documents that discuss the impact...*”. These twelve queries and their retrieval performance in MAP are listed in Table 3.6.

In next subsection, we will report our experimental results of above topics. We will evaluate the fuzzy models and graph-based models by using different IU similarities. The results reported in this chapter can be used to make comparison with the retrieval performance reported in next chapter.

Table 3.6 Topics with discourse type *impact* and their retrieval performance

Discourse Type: impact	
Topic ID and Title of Query	MAP
310 Radio Waves and Brain Cancer	0.084
345 Overseas Tobacco Sales	0.259
352 British Chunnel impact	0.205
391 R&D drug prices	0.105
407 poaching, wildlife preserves	0.358
448 ship losses	0.008
610 minimum wage adverse impact	0.051
641 Valdez wildlife marine life	0.421
645 software piracy	0.637
666 Thatcher resignation impact	0.005
678 joint custody impact	0.125
686 Argentina pegging dollar	0.491
Mean MAP	0.229

3.3.2 Experimental Results of Fuzzy Model

In this section, we will test the different weighting schemes proposed in our fuzzy models. For each topic, the original retrieved list returned 1000 documents by using the BM25 term weight of the 2-Poisson model [Robertson 94] using the standard parameter setting [Robertson 97] (i.e., $k1 = 1.2$ and $b = 0.75$) with passage-based retrieval and pseudo relevance feedback (PRF). The performance of above original retrieval is used as baseline and similarity score obtained from above model is used for later document re-ranking. For each document, we use the similarity score obtained by formula 3.1 to re-weight the document by multiplying with the original similarity score, see the formula beneath. To use multiplication to combine the two score is because we find it can lead to better performance than summation and many fuzzy AND operations.

$$S' = S_0 \cdot sim(doc, q) = S_0 \cdot \frac{1}{tf(doc)} \cdot \sum_{i=1}^{tf(doc)} sim(IU_i(doc), q)$$

In above formula, S' is the document score for re-ranking, S_0 is the original similarity score of the baseline and $tf(doc)$ is total number of topic entity terms in the document doc , which is equal to the number of Ius belonging to the document doc . We use arithmetic average to derive the aggregation function agg_i in formula 3.1 because it's simple and already provides fairly good performance. We noticed that the results of [Wu 2007] are consistent with ours in that to use average arithmetic to aggregate the local similarity measures (or context scores, IU score in our work) is better than extended Boolean AND operator and Dombi intersection operator under various

parameter settings. [Wu 2007] also tried to discover the best aggregation operator to combine the context scores in a document. They found that when $p=1$, the result of extended Boolean AND is the best among other setting of p ($p=5, 10, 20, 40, \infty$). It is also better than Dombi operator with different values of p ($p=1, 5, 10, 20, 40, \infty$).

In Table 3.7, we report the mean MAP of the topics belonging to each discourse types by the constant weighting schemes. We note that the results based on “**Ortf**” are same as the baseline results. It’s because each IU contains at least one of the topic entity terms and the “**Ortf**” similarity score is one for all the Ius. So the above re-ranking formula is degenerated into $S'=S_0$. In terms of mean MAP of all the topics, “**SUMtf**” is the best measure. These weighting schemes do not perform consistently on different discourse types since we can see that the best result for each discourse type may result from different weighting scheme. We report the results from term feature based weighting schemes in Table 3.8. **idf** is only based on term specialty which is measured by inverse document frequency. **SP** is only based on the distance between the term and the centre term (locates in the middle) of IU. And **IDF-POST** is based on both term specialty and the distance. A simple distance measure is used for **SP** and **IDF-POST** where the constant is set to 0.001, see the following formula. The difference between p and w is the distance between the weighted term at the p -th position and the middle term in the IU in terms.

$$f_p(|p-w|) = \frac{1}{|p-w+0.001|} \cdot \frac{1}{1 + \frac{1}{|p-w+0.001|}}$$

From the result of Table 3.8, we find that **SP** is the best one compared with the other two that consider idf of the term. **SP**’s performance is consistent among different

discourse types.

Table 3.7 Results in MAP of fuzzy model based on constant weighting schemes

Discourse Type	Number of topics	baseline	Ortf	ANDtf	SUMtf	dtf	MINtf
adv./disadv.	8	0.2086	0.2086	0.2136	0.2245	0.2221*	0.2149
country	6	0.1952	0.1952	0.2002	0.2027	0.2042*	0.2002
Reason	9	0.2241	0.2241	0.2373*	0.2336	0.2339*	0.2377
Impact	12	0.2291	0.2291	0.2341	0.2338	0.2315*	0.2348
mean		0.2173	0.2173	0.2244	0.2263	0.2253	0.2251

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Table 3.8 Result in MAP of fuzzy model based on term features based weighting schemes

Discourse Type	Number of topics	baseline	idf	SP	IDF-POST
adv./disadv.	8	0.2086	0.2179	0.2216*	0.2171
country	6	0.1952	0.1923	0.2033*	0.1927
reason	9	0.2241	0.2277	0.2304*	0.2239
impact	12	0.2291	0.2329	0.2343*	0.2333
mean		0.2173	0.2212	0.2251	0.2202

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval .

In order to discover whether the retrieval performance of our fuzzy model depends on the ways of measuring the term distance, we propose different functions to measure the distance and they are listed in Table 3.9. They are all monotonically decreasing functions since we assume that we should sign a term smaller weight if it occurs far way from the middle of IU. Two of the functions are exponential, two of them are polynomial and one function is linear. Different types of functions are used to provide different decreasing rates. The proposed functions also include two strictly

convex and two non-convex functions. Based on them, we will study the performance of using the functions with the property that function value decreases more quickly at the small arguments (f_4 and f_5) and function value decreases more quickly at the big arguments (f_1 and f_2). In these functions, p is the function argument indicating the position of a term in IU. W is a constant which is the position of the middle of an IU and it is also equal to the half width of the IU. (The size of IU is $2w+1$). In specific, we set the constant r to 2 and α to 1.5 and we draw the graphs of the five functions with discrete points at each integer between 1 and 20. (The IU size is 41 so the maximum distance from a term to the middle term of IU is 20) Figure 3.3 shows the graphs of the five functions and we can see the different decreasing trends of the function value with the increase of the argument, which denotes the distance between a term and the middle of IU.

Table 3.9 Five distance measuring functions

	Convex	Function Type
$f_1 = 1 - \alpha^{(p-w)}, (a > 1)$	non-convex	exponential function
$f_2 = \sqrt[r]{1 - (p/w)^r}, (r > 1)$	non-convex	polynomial
$f_3 = 1 - (p-1)/(w-1)$	convex	linear
$f_4 = 1 - \sqrt[r]{1 - (\frac{p}{w} - 1)^r}$	strictly convex	polynomial
$f_5 = \alpha^{(w-p)}, (a > 1)$	strictly convex	exponential function

Figure 3.3 Graph of the five distance measuring functions

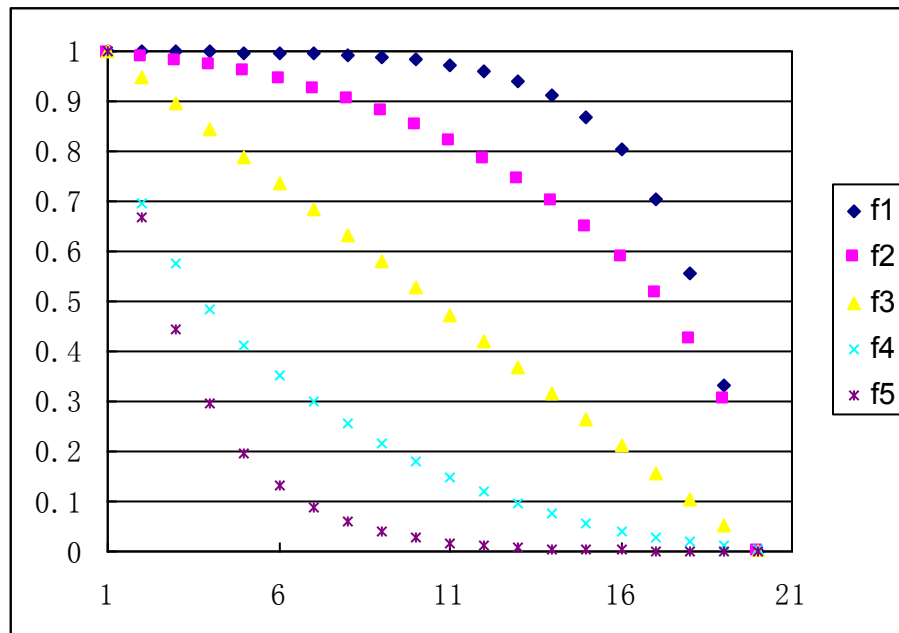


Table 3.10 Result in MAP of SP based on different distance measuring functions

Discourse Type	Number of topics	Baseline	SP with f1	SP with f2	SP with f3	SP with f4	SP with f5
adv./disadv.	8	0.2086	0.2254*	0.2236	0.2236	0.2228	0.2223
country	6	0.1952	0.2023*	0.2020	0.2020	0.2030	0.2030
reason	9	0.2241	0.2306*	0.2307*	0.2309	0.2308	0.2304
impact	12	0.2291	0.2353*	0.2353*	0.2353*	0.2342	0.2337
mean		0.2173	0.2262	0.2257	0.2258	0.2254	0.2250

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Among all IU similarity measures in the fuzzy model, there are two measures considering the distance between the topic entity term that is to be weighted and the middle of IU: **SP** and **IDF-POST**. In Table 3.8, their retrieval performances are presented and we know that these results came from the simple distance measuring formula. We are interested in whether different distance measuring functions can improve the retrieval performance after re-ranking. Therefore, we utilize the five

functions proposed above in the calculation of these two similarity measures. We report the results of **SP** in Table 3.10 and **IDF-POST** in 3.11. Generally speaking, from function f_1 , f_2 to f_5 , the retrieval performance is decreasing for the two distance-based similarity measures (**SP** and **IDF-POST**). Based on f_1 , **SP** has the performance 0.2262 in mean MAP for all the topics in the four discourse types and **IDF-POST** has the performance 0.2218. We find that the best performance of constant weightings on all topics is **dtf** and the mean MAP is 0.2263. Compared with **dtf**, the similarity measure that does not use the term position information, the best result of **SP** (with the function f_1) has the mean MAP performance 0.2262. So in our fuzzy models based on IU, term position information cannot help to improve the retrieval performance. We will study whether term position information can improve the retrieval for graph-based models in the later section.

Table 3.11 Result in MAP of IDF-POST based on different distance measuring functions

Discourse Type	Number of topics	Baseline	IDF-Post with f1	IDF-Post with f2	IDF-Post with f3	IDF-Post with f4	IDF-Post with f5
adv./disadv.	8	0.2086	0.2220	0.2213	0.2213	0.2190	0.2179
country	6	0.1952	0.1920	0.1920	0.1920	0.1927	0.1923
reason	9	0.2241	0.2247	0.2247	0.2250	0.2237	0.2240
impact	12	0.2291	0.2343	0.2343	0.2343	0.2339	0.2328
mean		0.2173	0.2218	0.2216	0.2217	0.2208	0.2202

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Finally, we report the performance of the re-ranking retrieval results by using fuzzy set based similarity measures. We use Dombi's conjunction (AND) operations in

our experiments. The conjunction operation of two numbers a and b is calculated by the following formula:

$$i_l(a, b) = \frac{1}{1 + \left[\left(\frac{1}{a + \alpha} - 1 \right)^l + \left(\frac{1}{b + \alpha} - 1 \right)^l \right]^{\frac{1}{l}}}$$

Note that this function differs from the standard Dombi's conjunction in that we bring in a small constant α to avoid zero division. Also, we notice that Dombi's conjunction has a parameter l to change its hard/soft level of the conjunction. We use this operator to combine the weights of each topic entity terms occurring in an IU. We report the performance in mean MAP by using different l values and the results are shown in Table 3.12. For each distinct topic entity term, we use its term frequency as its weight. After normalization, we combine the weights of all topic entity terms by using Dombi's intersection operation.

Table 3.12 Results in MAP by using Dombi intersection operation with different parameters

Discourse Type	Number of topics	baseline	$l =$					
			1	2	5	10	20	40
adv./disadv.	8	0.2086	0.2194	0.2173	0.2189	0.2193*	0.2193*	0.2191
country	6	0.1952	0.2078	0.2055	0.2040	0.2037	0.2038*	0.2038*
reason	9	0.2241	0.2419	0.2420	0.2426	0.2437*	0.2436*	0.2438*
impact	12	0.2291	0.2184	0.2205	0.2211	0.2209	0.2210	0.2209
mean		0.2173	0.2229	0.2227	0.2232	0.2234	0.2235	0.2234

“*” indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

The results in mean MAP by using Dombi intersection (AND) operation show that the $l=20$ produces the best performance. [Wu 2007] also stated that Dombi intersection

operation gives the best performance when p is set to 20. (Note that their parameter p is just the parameter l in above formula). So our work is consistent with theirs in the Dombi parameter that gives the best performance and both of the work use Dombi intersection as aggregating function to combine local relevance information. The difference is they used Dombi intersection to combine the context scores for a document and we use Dombi intersection to combine the term weights (scores) for an IU.

In conclusion, re-ranking based on fuzzy models with three types of term weighting schemes (constant term weighting, feature-based term weighting and fuzzy set term weighting) can enhance the original retrieval performance. Among these three types of term weighting, constant term weighting contributes the best results with **SUMtf** (mean MAP is 0.2263). Different distance measuring function can affect the results of feature-based term weighing if distance factor is considered. The best result (mean MAP is 0.2262) of **SP** (feature-based term weighing based on distance factor only) is very close to **SUMtf** and this result is based on a non-convex exponential function $f_1 = 1 - \alpha^{(p-w)}$, ($a > 1$). The best results of re-ranking based on fuzzy set term weighting are better than the original results but worse than the two results mentioned above.

3.3.3 Experimental Results of Graph-based Model

In this section, we will present the experimental results of the graph-based models. For

each topic, the original retrieved list also contains 1000 documents retrieved by using the BM25 term weight of the 2-Poisson model [Robertson 94] using the standard parameter setting [Robertson 97] (i.e., $k1 = 1.2$ and $b = 0.75$) with passage-based retrieval and pseudo relevance feedback (PRF). The performance of above original retrieval is used as baseline and similarity scores obtained from graph-based models are used for later document re-ranking. As the method we re-rank the retrieved documents depending on the fuzzy set models, we use the similarity scores obtained by formula 3.4 to re-weight the documents by multiplying with the original similarity score. We also use multiplication to combine the two scores because it brings better performance than other operators such summation or fuzzy set AND operations. The re-ranking formula is:

$$\begin{aligned}
S' &= S_0 \cdot sim(doc, q) = S_0 \cdot \frac{1}{tf(doc)} \cdot \sum_{i=1}^{tf(doc)} sim(IU_i(doc), q) \\
&= S_0 \cdot \frac{1}{tf(doc)} \cdot \sum_{i=1}^{tf(doc)} f(G(V_i, E_i))
\end{aligned}$$

In above formula, S' is the document score after re-ranking, S_0 is the original similarity score of the baseline and $tf(doc)$ is total number of topic entity terms in the document doc which is equal to the number of Ius belonging to the document doc . We use arithmetic average to aggregate the scores of all Ius in document doc as we did for the re-ranking based on the fuzzy models. $G(V_i, E_i)$ is the graph representing the i -th IU and $f()$ is a function to calculate the similarity score of this IU based on a certain graph-based method.

First, we present the retrieval performance of graph-based model using allo-T edges with **at01** edge selection scheme, in which all allo-T edges are considered in the calculation of the IU similarity score. For each allo-T edge, three term features can be

taken into account for computing the edge relevant evidence score: distance in words between the two terms in the IU, term specialty and order of the two terms. In first half of Table 3.13, the “const” column shows the results obtained by assigning each allo-T edge a constant weight; “dis” column shows the results of considering term distance factor; “ord” column shows the results of considering term order factor and “dis+ord” column shows the results of considering both term distance and term order.

We will introduce the four types of edge weighting methods in details. The constant weighting (“const”) is to assign each allo-T edge an constant weight based on the following formula: $\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = C$. In the experiments, we set the constant C by 1 because this value does not affect the ranks of the documents. In this kind of weighting, different allo-T edges are weighted equally; no matter what they link are different pairs of terms (e.g. “health”-“care” and “health”-“Britain”) or same pairs of terms occurring at different positions in text (e.g. the first “health” in the IU- the first “care” and the first “health”- the second “care”). The constant weighting can be used as a baseline of each selection scheme for us to check whether considering the features of terms (e.g. specialty, distance, etc.) enhances the performance.

“dis” weighting is according to the following formula to compute the relevant evidence score for an edge, which only considers the factor of distance between two terms linked by the edge:

$$\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = f_p(|p - q|) = \frac{1}{1 + \frac{1}{|p - q + \alpha|}}, \quad (\alpha \neq 0)$$

The second subscripts p and q of term t are the positions of the terms in the IU. The difference between p and q is the distance in word of the two terms. So we can build a

function to take distance as the argument. In the results shown in Table 3.13, the simple position based weighting (**SP**) in the fuzzy model is adopted in above formula to bring in the distance factor for the calculation of the relevant evidence score of an edge. We have used the same distance measuring function to compute the IU similarity **SP** which belongs to feature-based weighting methods in fuzzy model. So we also use this distance function in order to make comparison between the different models possible. By comparing the results of “dis” with “const”, we can conclude that bringing in the distance factor cannot improve the graph-based model based on allo-T edges with **at01** edge selection scheme. In addition, we can draw the same conclusion by comparing “ord” with “dis+ord”, the distance factor is considered for the latter one.

“ord” weighting is based on the following formula to compute the relevant evidence score for an edge, which only considers the term order factor (occurrence sequence) of the two terms linked by the edge:

$$\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = \text{order}(t_{i,p}, t_{j,q}) = \begin{cases} a, & (i-j) \cdot (p-q) > 0 \\ b, & (i-j) \cdot (p-q) < 0 \end{cases} \quad (a > b)$$

The first subscript of a term t is the position (or sequence number) of the term occurring in the title part of a topic (title query) and second subscript is the position of term t in the IU. If the two terms linked by an edge occur in the IU with the same order as they occur in the title part of the topic (i.e. $(i-j)(p-q) > 0$), we assign a larger weigh a for this edge; otherwise, we assign a smaller weight b for it. This kind of weighting is based on the assumption that if two terms that occur with the same order as they occur in the query, they are more likely to present the relevant information of the query. Obviously we know that term order greatly make senses if the two terms can compose a phrase or commonly used expression. Let us take TREC query topic No. 608 as

example, “*social*” and “*security*” can compose an often-used phrase when the two terms occur like this “*social security*” and they cannot compose one if they occur in the reverse order. However, the topic entity term “*social*” and “*tax*” do not have a serious problem of term order. Obviously considering the order of some certain topic entity terms is better than consider all. However, in order to avoid the human intervention, we consider the term order factor for all the terms in the topic. In the experiments, a is set to 2 and b is set to 1. Since the relevant evidence scores of the edges are usually aggregated by summation and the IU scores are also aggregated by summation, so if the ratio of a to b is a constant, the ranking of the documents remains the same. The ratio in original setting is 2 and we can tune this ratio later if necessary.

Table 3.13 Results in MAP by using allo-T edges with at01 edge selection

Discourse Type	Number of topics	baseline	at01			
			const	dis	ord	dis+ord
adv./disadv.	8	0.2086	0.2181	0.2221	0.2248*	0.2244
Country	6	0.1952	0.2075	0.2018	0.2063*	0.2025
Reason	9	0.2241	0.2386	0.2307	0.2412*	0.2351
Impact	12	0.2291	0.2370	0.2368	0.2358*	0.2350
mean		0.2173	0.2280	0.2259	0.2296	0.2270

Discourse Type	Number of topics	baseline	at01			
			idf	idf+dis	idf+ord	idf+dis+ord
adv./disadv.	8	0.2086	0.1794	0.1754	0.1854	0.1811
country	6	0.1952	0.1942	0.1913	0.1967	0.1932
reason	9	0.2241	0.2196	0.2164	0.2214	0.2180
impact	12	0.2291	0.2300	0.2290	0.2332	0.2304
mean		0.2173	0.2096	0.2071	0.2130	0.2096

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Results of “dis+order” column come from combining the two factors:

$$\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = f_p(|p - q|) \cdot \text{order}(t_{i,p}, t_{j,q})$$

We use multiplication to combine the two factors because we need a conjunctive operator. Compared with “const”, “ord” can outperform “const” on average and for most of the topics “ord” produces better results than “const”. It shows that bringing the term order factor can slightly enhance the performance of graph-based model. We can draw the same conclusion by comparing results of “dis” with the results of “dis+ord”.

After reviewing the results of the second half of Table 3.13, we will see the results of bringing in the specialty factor for the graph-based model based on allo-T edge with **at01** edge selection scheme. Results of “idf” column are based on the following formula:

$$\varepsilon(\text{edge}(t_{i,p}, t_{j,q})) = IDF(t_i, t_j) = \text{idf}(t_i) \cdot \text{idf}(t_j) = \left[1 - \frac{\lg df(t_i)}{\lg N}\right] \cdot \left[1 - \frac{\lg df(t_j)}{\lg N}\right]$$

We know that “idf” only brings in the specialty factor for computing the relevant evidence score and it does not consider the positions of the terms linked by an edge or their occurrence sequence. The results of the four columns in the second half of Table 3.13 are obtained by bringing the “idf” factor into the four columns in the first half of Table 3.13. For example, results of “idf+dis” are based on the following formula:

$$\begin{aligned} & \varepsilon(\text{edge}(t_{i,p}, t_{j,q})) \\ &= IDF(t_i, t_j) \cdot f_p(|p - q|) = \left[1 - \frac{\lg df(t_i)}{\lg N}\right] \cdot \left[1 - \frac{\lg df(t_j)}{\lg N}\right] \cdot \frac{1}{1 + \frac{1}{|p - q + \alpha|}}, \quad (\alpha \neq 0) \end{aligned}$$

By comparing the results of the second half of Table 3.13 with the first half, we find that bringing the specialty factor by using idf value hurts the performance. It is not

consistent with the traditional view of points on idf value. We think that this is perhaps due to the Ius that contain only one term. A term with very specific meaning has a large idf value. However, the basic unit of graph-based model based on allo-T edges is an edge that links two distinct topic entity terms. So the Ius that contain only one term cannot be counted no matter how specific the term is. So idf value is not so helpful to the graph-based model. In later experiments on the graph-based models, we find the similar phenomenon. So we do not report the results that consider the specialty factor.

Table 3.14 Results in MAP by using allo-T edges with at02 edge selection

Discourse Type	Number of topics	baseline	at02			
			const	dis	ord	dis+ord
adv./disadv.	8	0.2086	0.2154	0.2176	0.2209*	0.2231
Country	6	0.1952	0.2060	0.2002	0.2057	0.2020
Reason	9	0.2241	0.2372	0.2288	0.2397	0.2328
Impact	12	0.2291	0.2376	0.2368	0.2360	0.2351
mean		0.2173	0.2270	0.2241	0.2283	0.2261

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

We report the results of graph-based model based on allo-T edges with **at02** edge selection scheme in Table 3.14. The edge set that **at02** selects is a subset of the edge set selected by **at01**. It consists of the edges that link the center term of IU. If there is no allo-T edge linking with the center term in an IU graph, no edge is selected by this selection scheme; as a result, this IU is ignored when computing the similarity score of the document containing this IU. The results in “const”, “dis”, “ord” and “dis+ord” columns are based on the same methods with Table 3.13. They are based on different factors to calculate the relevant evidence score of an edge.

Table 3.15 Results in MAP by using allo-T edges with at03 edge selection

Discourse Type	Number of topics	baseline	at03			
			const	dis	ord	dis+ord
adv./disadv.	8	0.2086	0.2190	0.2203	0.2265*	0.2246
Country	6	0.1952	0.2055	0.2012	0.2058*	0.2032
Reason	9	0.2241	0.2344	0.2293	0.2379*	0.2350
Impact	12	0.2291	0.2366	0.2360	0.2356*	0.2355
mean		0.2173	0.2267	0.2247	0.2290	0.2273

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Table 3.15 shows the results of graph-based model based on allo-T edges with **at03** edge selection scheme. As **at02**, the edge set that **at03** selects is also a subset of the edge set selected by **at01**. For each possible pair of two different topic entity terms (the number of possible pairs is C_2^{DTF} , DTF is the number of distinct topic entity terms in an IU), **at03** selects the edge(s) linking the nearest two terms. The results in “const”, “dis”, “ord” and “dis+ord” columns are based on the same methods with above two tables. They are based on different factors to calculate the relevant evidence score of an edge. **At02** and **at03** are based on two different edge selection strategies: **at02** emphasizes the importance of the fact that the middle term is one of the two terms linked by an edge in evaluating the relevant evidence of the edge. And **at03** emphasizes the importance of the distance between two linked terms and select the shortest edge as the representative of all edge that linked the same terms. These two strategies can be evaluated by comparing the results of **at02** and **at03**.

By comparing **at02** with **at03**, the “const” column of **at02** is better, which shows the edges linking the middle terms (one of the topic entity terms) are important

to the relevance of IU if edges are equally weighted regardless of which terms they link. However, if edges are weighted differently and the features of linked terms are considered, the case is different. For the results of considering other term features (the other three columns), **at03** is better than **at02**, which shows if the features of terms (order and position) are considered, selecting the shortest edge is better than selecting the edges linking the middle term of IU. In conclusion, the first strategy that emphasizes the middle term outperforms when edges are equally weighted; if the features of two terms linked are considered when weighting an edge, the second strategy that selecting the shortest edge outperforms the first one.

Table 3.16 shows the results of graph-based model based on allo-T edges with **at04** edge selection scheme. The edge set that **at04** selects is the intersection set of the edge set selected by **at02** (linking the middle) and by **at03** (nearest). Among the C_2^{DTF} edges selected by **at03**, **at04** only keeps the edges that link with the middle term. If the number of distinct topic entity term in an IU is DTF , **at04** selects $DTF-1$ edges that link the middle term with each of the other topic entity terms. Comparing the results of **at04** with **at02**, **at03**, we find that the performance of **at04** lies between the results of **at02** and **at03**.

After comparing the results of different edge selection schemes, let us review different weighting methods inside each selection scheme. By comparing the results of the four columns “const”, “dis”, “ord” and “dis+ord” inside each table, we find that “ord” outperforms the other three. This is consistent for edge selection methods **at01**, **at02**, **at03** and **at04**. This can be easily explained when the edges that link two nearest topic entity terms are used (that is just what **at03** selects), the order of two terms is

important to the presentation of the relevant information because two adjacent terms with right occurrence order can compose a phrase which is more accurate and specific than single terms.

Table 3.16 Results in MAP by using allo-T edges with at04 edge selection

Discourse Type	Number of topics	baseline	at04			
			const	dis	ord	dis+ord
adv./disadv.	8	0.2086	0.2178	0.2185	0.2246*	0.2228
Country	6	0.1952	0.2053	0.2008	0.2060	0.2028
Reason	9	0.2241	0.2336	0.2276	0.2362*	0.2324
Impact	12	0.2291	0.2363	0.2360	0.2366	0.2355
mean		0.2173	0.2260	0.2238	0.2285	0.2262

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Table 3.17 Results in MAP by using allo-T edges with at05 edge selection

Discourse Type	Number of topics	baseline	at05			
			const	dis	ord	dis+ord
adv./disadv.	8	0.2086	0.2166	0.2161	0.2179	0.2173
Country	6	0.1952	0.2032	0.198	0.2107*	0.2055
Reason	9	0.2241	0.2321	0.2278	0.2347	0.2309
Impact	12	0.2291	0.2371	0.2368	0.2357	0.2367
mean		0.2173	0.2253	0.2231	0.2271	0.2254

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.9% confidence interval.

Among the different edge selection schemes based on “ord” weighting method, the results of **at03** are the best, which is based on the strategy that emphasizes the distance of the two linked term. Hence, it’s reasonable to investigate more on this strategy. So we propose **at05** as an extension of **at03**. The edge set that **at05** selects is

a subset of the edge set selected by **at03**. Among the C_2^{DTF} edges selected by **at03**, **at05** only keeps the shortest edges. That is to say that **at05** selects only one edge, which is the shortest among all the possible allo-T edges of an IU.

Table 3.17 shows the results of graph-based model based on allo-T edges with **at05** edge selection scheme. We find that **at05** is worse than the other four selection schemes. We think the reason is it uses too few edges (only one edge). However, it can still improve the original retrieval performance on average, which shows the importance of the existence of two different topic entity terms in determining the relevance of an IU.

In conclusion, to select all edges is better than other edge selection schemes that select part of edges. This is probably results from the small size of IU in our experiments. However, the other four edge selection schemes are a little worse than selecting all edges, which can reflects the importance of some special edges in an IU graph to the relevance of an IU. Hence we think that edge selection schemes that select part of edges according to some criterions are still potential, especially when IU size is big. As for the calculation of edge relevant evidence, “ord” produces the best results for all of the five edge selection schemes. It shows that term order is more important than them distance between two terms or term specialty in quantifying the relevance of an edge that links two terms.

Summary

As for IU-based retrieval, the simple weighting scheme (e.g. SUMtf is just the

summation of the term frequencies in an IU) is comparable with other complicated weighting schemes (considering a pair of terms, order of two different terms, and position of a term in the IU etc.). Since all the terms in an IU have close syntactic, semantic and pragmatic relations with the centre term and the centre term is a topic entity term appearing in title query of a topic, any occurrence of any topic entity term in an IU determines the relevance of an IU to a topic to a great extent.

Among the features we studied, the distance between a term and the centre of an IU is more influential than other schemes including idf value of the term, which shows the philosophies of IU-based retrieval (or context window). Also, the simple weighting scheme is better than the fuzzy-set weighting schemes. This conclusion is consistent with and justifies the effectiveness of the simple weighting scheme for discourse type related linguistic sequences in Chapter 4.

The IU-based retrieval is able to outperform the traditional 2-position model with BM25 weighting scheme for the TREC topics within the three discourse types we study. The retrievals do not consider using discourse type information, which provides us good baselines for the later comparison between using and not using discourse type related information.

CHAPTER 4

DISCOURSE TYPE BASED INFORMATION RETRIEVAL

In this chapter, we propose the concept of “discourse type” of a topic in order to accurately knowing what an information need is asking for. Different types of linguistic sequences are used to further enhance the poor retrieval performance caused by the diversity and complexity of information need. We also deeply investigate the TREC topics and determine the appropriate discourse types to support our study. We also study the characters of different types of linguistic sequences in the application of discourse type based information retrieval. The experimental results of discourse type based retrieval are also generated from IU-based retrieval, which can reasonable use the results of Chapter 3 as baseline for comparison and analysis.

4.1 Introduction of Discourse Type

The effectiveness of information retrieval (IR) systems varies substantially from one topic to another. This may be due to the diversity of the user information need and the

diversity also implies that some information need is very complicated to express. Common retrieval systems cannot perform well for all the different kinds of topics since they still deploy a relatively simple representation of information need and pay little special attention to from what perspective or aspect (our so-called discourse type, such as properties, relations) to approach the topic subject can really satisfy the information need. Potentially, to solve the diversity of user information need may be one of the basic problems in IR.

The fundamental premise of the investigation of the diverse information need is that the information need must be stated as clearly as possible. We use TREC Robust Track topics [Voorhees 05] in our study because it is generally accepted that these topics include a clear statement of what criteria make a document relevant. The format of a TREC Robust Track topic statement has been stable since TREC-5. It is well acknowledged that a topic statement generally consists of four sections: an identifier, a title, a description and a narrative. The title field consists of up to five words that best describe the topic. The description field is a one sentence description of the topic area. The narrative gives a concise description of what makes a document relevant.

Among the title, description, and narrative sections of the query, we study the information need of a topic mainly based on the description section because

(1) The title section is often too short to completely present the information need. For example, the title section of topic 699 is “term limits” and it does not provide the necessary information--- pros and cons, which is stated in the description section, see Table 4.1.1. This necessary information is important for determining whether a document is relevant or not. Another example is topic 700

and its title section is “gasoline tax U.S.”. Based on the title section, we may guess that this topic probably requires information on how much is the tax on gasoline in the U.S.. We know it’s not true according to the description section, and this topic is looking for information on “What are the arguments for and against an increase in gasoline taxes in the U.S.?”

(2) The narrative section contains several sentences stating under what condition a document is relevant or irrelevant. However, the narrative sections lack of a consistent format which makes it difficult to induce a general and consistent framework for all the topics. For example, the narrative sections of almost all the topics provide the condition or suggest some special content for a document to be relevant. However, not all narrative sections provide the condition for a document to be irrelevant.

(3) Through our study, we find that almost all the description sections of all topics provides clear enough information about an information need, which usually covers two types of information: one is on an independent topic entity (such as “term limits” in topic 699) and another one is called “discourse type” by us which are the functions, including properties and relations, of above independent topic entity. These subordinate functions always cannot exist independently, such as “pros and cons” in topic 699 and “argument for and against” in topic 700. We should mention that description sections of some topics (e.g. topic ID 361, 368, 379, 384, 390) appearing in TREC-7 do not provide discourse type information and their narrative sections suggest some discourse types. Each of these topics contains several discourse types in the narrative sections and we do not study them at the

moment since we study the topics having only one discourse type.

Table 4.1.1 Examples of the topics that own the same discourse type

Discourse Type	Topic ID	Description section of query
pros and cons	605	What are the pros and cons of Great Britain's universal health care system?
	624	What are the pros and cons of developing the Strategic Defense Initiative (SDI) also known as "Star Wars"?
	699	What are the pros and cons of term limits?
arguments for and against	621	What are the arguments for and against Great Britain's approval of women being ordained as Church of England priests?
	635	What are the arguments for and against doctor assisted suicide in the U.S.?
	700	What are the arguments for and against an increase in gasoline taxes in the U.S.?

By studying the 250 topics of TREC Robust Track 2004 (including topics of TREC-6, TREC-7, TREC-8 and 100 new queries, all in English), we found that the discourse type can be easily and consistently detected from the description section of the topic using some cue phrases (e.g., “pros and cons” and “arguments for and against”). Some topics have more than one discourse types, such as “frequency and cause” of topic 336, “quantity and country name” of topic 414. We don’t study the topics having more than one discourse types at the moment because these topics are more complex and only account for less than 14% of all the topics. More importantly, we found that the topics having only one discourse type share some common discourse types. For example, the description sections of TREC topic 605, 624 and 699 show that the three queries are looking for pros and cons of some policies or projects. The topic 621, 635 and 700 are all enquiring about the arguments for and against an event or

phenomenon. See Table 4.1.1.

Table 4.1.2 The discourse types discovered from TREC topics

Discourse Type of Topic	Num of Topics
advantage/disadvantage	8
any information	41
approach	2
argue	3
ban	2
benefit	3
commercial use	2
complex queries (need more than one discourse types)	39
country	6
drug	2
effect	4
effort	2
evident	2
extent	3
incident	2
impact	12
method	3
number	4
other groups (containing only one topic)	75
place	2
procedure or process	3
reason	9
relation	2
research	2
role	2
status	2
step	4
treatment	2
use	4
ways	2

In order to quantify the problem of the diversity of information need, we manually grouped 250 TREC Robust Track topics based on the discourse type. In each group, the discourse types of the topics are same or very alike. For example, above topic 605, 624, 699 are grouped together because their discourse types are same. We also add the topics whose discourse types are “advantages and/or disadvantages” into the group containing topics whose discourse type is “pros and cons” because these two discourse types are very alike. The reasonableness of combining topics whose discourse types are “very alike” can be justified by the similarity of their relevant documents in our experiments. Since the grouping is based on the explicit cue phrases appearing in the description sections, the repeatability of grouping queries can be ensured. The total number of topic studied is 249 since one topic has not any relevant documents.

From Table 4.1.2, we can find there are 39 topics (queries) that require more than one discourse types (e.g. “history and extent”, “roots and prevalence”) and each of these topics has a unique discourse type. We group them into “complex queries”. There are 41 topics that do not show a mandatory discourse type and we group them into “any information”. The rest topics can be grouped according to their discourse types from their queries’ description. However, most of the discourse type groups contain very few (less than 4) topics. For example, there are 75 discourse type groups containing only one topic so that we do not explicitly show their discourse type and group them into “other groups” for brevity. Obviously, if we intend to find the common potential linguistic expressions or structures to present a discourse type, it is unreasonable to use the these three groups “complex queries”, “any information” and

“other groups” into our research since we need to discover and collect the generality from different topics that belongs to one discourse type group rather than one single topic with its own discourse type. Theoretically, we need to use the groups that contain more topics to justify the prevalence of the conclusions. Empirically, we need more training data to cover the complicated phenomena that could appear in natural language. Finally, we choose “advantage/disadvantage”, “reason” and “impact” as examples because they contains more topics compared with other discourse type groups.

In this thesis, we don't focus on how to automatically recognize the discourse type of a topic because (a) In practice, the user can explicitly indicate the discourse type she/he wants by adding a command to the query just like Google command “define: ”. (b) If the user cannot indicate what discourse type(s) she/he wants, we can prompt her/him by producing multiple retrieval lists where each retrieval list is the retrieval of the topic plus an assumed discourse type. (c) It is easy to recognize the discourse type if the information need is written like TREC “description” section by using simple natural language processing techniques.

We interest in how to recognize the discourse type of text in documents. We all know that a document is a big language unit so that a document may contain several discourse types as well as many topic entities. The judgement of the relevance of document adopted by TREC is based on the disjunctive relevance decision (DRD) principle [Kwong 04] which states that any part of a document that is considered relevant implies that the whole document is relevant. So we first decompose a document into some small parts in order to assume that each small part only covers

one topic entity and has one discourse type. If one part among these small parts is talking about the required topic entity and also has the required discourse type, the document is relevant. There are many possible parts can be extracted from a document and obviously it's more efficient to process the parts that are very likely to cover the topic entity than process all possible parts. Hence, we put forward the concept of "information unit".

For a document and a topic, Information unit (IU) is a fixed size text window extracted from the document and the centre term of the text window is one of the entity terms of the topic. The entity terms of the topic are selected from the terms in the title section because the title section contains terms that best describe the information need and, also, it's possible to compare our work with other's work using title section. The terms in title section contains the terms related with the topic entity, called topic entity terms, and some title sections also contains the terms relate with discourse type. For example, title section of topic 690 is "college education advantage", where terms "college" and "education" are topic entity terms and "advantage" is obviously related with the discourse type. In query "term limits"(title section of topic 699), "term" and "limits" are both entity terms and the discourse type related terms of this topic (i.e. "pros and cons") are included in the description section. For a document and a topic, an IU extracted from the document is a part of the document that contain at least one topic entity term in the centre. Hence, we assume that all the IUs are talking about the topic entity. Now we deduce our work from determining whether a document is relevant or not to determining whether the discourse type of an IU match the discourse type of topic.

We find that a discourse type usually has its individual ways to represent itself in text and it often uses some characteristic words, phrases or sentence patterns to convey some special meanings. Let us take discourse type “advantage and disadvantage” as example. A direct expression of “advantage and disadvantage” of an entity can depend on talking about its ability or inability, comparison with the alternatives, or measures related with money (e.g. *expensive*) or time (e.g. *efficient*). Indirect expression can depend on people’s opinion on it (e.g. *accept/reject*, *support/oppose*), effect and influence of the entity (e.g. *harm/help*). The frequent occurrence of words having above meanings have been noticed not only by manually checking the text talking about “advantage and disadvantage” but also by the statistics on the IUs extracted from relevant documents of the queries with this discourse type.

The simplest way to retrieve documents by considering discourse type is to directly add the name of target discourse type into the query. For example, we can perform the query expansion for the nine topics of discourse type reason (see Table 3.5) by adding “reason” into the original queries. As a result, the topic No. 333 “Antibiotics Bacteria Disease” becomes “Antibiotics Bacteria Disease reason”. In the actual web search, Internet users may use this way to search for the specific relevant information. However, our experiments show that this is not an effective way.

To directly add discourse type terms in the query cannot generally improve retrieval performance; sometimes even severely degrade the retrieval performance. In Table 4.1.3 (A, B and C), columns “original” show the baseline results used in our thesis and columns “adding” shows the results after several discourse types terms are added in the original queries. However, we lost the records of which discourse type

terms were used. Therefore, as compensation, we did more experiments on the retrievals using original queries and the queries added with discourse term. The experiments were performed on a different version of the same search engine with the same parameter setting and the results are shown in Table A.4.1 in Appendix 4 at the end of this thesis. The baseline (“original” columns in Table A.4.1) in the new experiments is a little different with the baseline (“original” columns in Table 4.1.3) in Table 4.1.3. But the comparison between the “original” and “adding” columns inside Table A.4.1 still makes sense and supports our conclusion that directly adding discourse terms into queries cannot generally improve the retrieval performance.

Table 4.1.3A Retrieval performance after adding the discourse type terms into the queries

Discourse Type: advantage/disadvantage		
Topic ID and Title of Query	Original	Adding
308 Implant Dentistry	0.124	0.076
605 Great Britain health care	0.176	0.126
608 taxing social security	0.133	0.104
624 SDI Star Wars	0.299	0.265
637 human growth hormone (HGH)	0.396	0.332
654 same-sex schools	0.041	0.024
690 college education advantage	0.004	0.003
699 term limits	0.496	0.418
Mean MAP	0.209	0.169

There are two reasons for the performance damage. First, apparently, most of the time discourse type information is not explicitly expressed in the documents as we mentioned before. A writer can claim a reason after connectives “because”, “due to” but not after “The reason is ...”. A reason can also be claimed without any connectives. So adding a discourse type name in the query does not significantly help recall. Furthermore, without a reasonable mechanism to deal with the discourse type terms,

adding such a term in the query will further damage the retrieval by lifting the documents containing this terms in the retrieved list, which will decrease the retrieval precision.

Table 4.1.3B Retrieval performance after adding the discourse type terms into the queries

Discourse Type: reason		
Topic ID and Title of Query	Original	Adding
333 Antibiotics Bacteria Disease	0.386	0.323
397 automobile recalls	0.481	0.411
436 railway accidents	0.156	0.114
628 U.S. invasion of Panama	0.224	0.198
636 jury duty exemptions	0.185	0.124
639 consumer on-line shopping	0.242	0.204
669 Islamic Revolution	0.052	0.042
670 U.S. elections apathy	0.187	0.135
673 Soviet withdrawal Afghanistan	0.104	0.087
Mean MAP	0.224	0.182

Table 4.1.3C Retrieval performance after adding the discourse type terms into the queries

Discourse Type: impact		
Topic ID and Title of Query	Original	Adding
310 Radio Waves and Brain Cancer	0.084	0.014
345 Overseas Tobacco Sales	0.259	0.217
352 British Chunnel impact	0.205	0.165
391 R&D drug prices	0.105	0.054
407 poaching, wildlife preserves	0.358	0.321
448 ship losses	0.008	0.006
610 minimum wage adverse impact	0.051	0.023
641 Valdez wildlife marine life	0.421	0.387
645 software piracy	0.637	0.584
666 Thatcher resignation impact	0.005	0.002
678 joint custody impact	0.125	0.052
686 Argentina pegging dollar	0.491	0.432
Mean MAP	0.229	0.188

In section 4.2 we formulate and evaluate features of word sequences and introduce the discourse type based retrieval based on word sequences. In section 4.3, we formulate and evaluate features of POS tag and introduce the discourse type based retrieval based on POS tag sequences. We formulate and evaluate features of word-POS tag sequences and introduce discourse type based retrieval based on word-POS tag sequences in section 4.4. The word-POS tag sequence discourse type model proposed in section 4.4 is hybrid by using the sequences composed of word and POS tags. For above retrievals, we report the retrospective and validation experiments in each section. In section 4.5, we make comparison among different types of sequences and attempt to find some ways to predict the performance of a sequence type.

4.2 Discourse Type Based Retrieval by using Word Sequences

Most traditional information retrieval models, e.g. vector space model, are based on the weights of individual word, including the fuzzy models we proposed in Chapter 3. In this section, we investigate and evaluate the word sequences that consist of at least two adjacent words and how these word sequences influence the discourse type based retrieval.

Table 4.2.1 An example of an IU and the extracted word bigrams and trigrams

690 LA031589-0075 the promise of economic growth and jobs , but also the chance to develop a unique society of cultural , <<educational>> , technological and artistic diversity . To do so , California must invest now in the necessary public support	
Word bigrams: the promise promise of of economic economic growth growth and ... the necessary necessary public public support	Word trigrams: the promise of promise of economic of economic growth economic growth and growth and jobs ... the necessary public necessary public support

The word sequences are called word bigrams, word trigrams and word N-grams if they consist of two words, three words and N words respectively. For example, Table 4.2.1 illustrates an IU extracted from document with DOCID LA031589-0075, which is the first IU of the retrieved documents of query 690. The centre term of the IU is “educational”, which is between “<<” and “>>”. We also show some of the word bigrams and word trigrams extracted from this IU. Some machine learning techniques are used to mine the word sequences with different lengths from the relevant documents in response to each discourse type.

Word sequence discourse type model is formulated to calculate the probability that the discourse type of a part of text (we also use information unit as the basic unit of a document) matches the desirable discourse type based on the word sequences of different lengths (word bigrams, word trigrams, etc.) that occur in this part of text. A word sequence discourse type model can be formulated as:

$$rel(doc, q) = \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))] \quad (4.2.1)$$

In formula 4.2.1, $rel()$ denotes the relevance between a given document doc and query q , $sim()$ denotes the similarity between an IU and a query based on non-discourse type methods, such as vector space model or IU similarity model. IU_i denotes the i -th IU of doc . agg is aggregating function to combine the measures obtain from all IUs of a document. $H()$ denotes the probability of IU has the discourse type dt based on the learned function. “ \wedge ” denotes a conjunctive function. A word sequence discourse type model based on the word sequence with a certain length has the factor $H()$ derived as:

$$H(IU_i, dt) = f_{dt}[ws(IU_i)] \quad (4.2.2)$$

$ws(IU_i)$ is the data structure consisting of all the word sequences with a certain length occurring in IU and their frequencies and positions. It's possible for a word bigram to occur more than once in an IU so frequency and position information of an IU is necessary to completely present an IU. And $f_{dt}[]$ is a function obtained from machine learning methods, which determines the probability of the discourse type of an IU presented by a word bigram data structure meets the desirable discourse type.

A discourse type is always explicitly expressed in text by some certain phrases and words, which is the premise of word sequence discourse type model. We have found that the frequencies of some certain phrases and words are much higher than others in relevant documents of the topics having the same discourse type. That is to say, if two topics have the same discourse type, the relevant documents of the two topics contain some common phrases and words that cannot be easily found in the documents other than the relevant documents of this type of topics. The word sequence discourse type model makes use of these special phrases and words to determine the probability of a

discourse type of an IU.

4.2.1 The distribution of word sequences in relevant documents

There are many measures that can indicate the distribution of a word sequence in the retrieved documents. In this study, we are also using IUs extracted from documents rather than the whole documents. This is based on the assumption that the phrases and words that express discourse type are more likely to appear near the topic entity terms. Since there are always many relevant and irrelevant documents in the retrieved list and each of the relevant and irrelevant documents contributes some IUs, when we evaluate the distribution of a word sequence, we need to consider how this word sequence distributes in each IUs of the relevant documents and irrelevant documents and then aggregate all the information together to have a measurement for the distribution of a word sequence in a retrieved list of a topic (e.g. a TREC query).

Let us take word bigram as an example. Some word bigrams occur frequently in both relevant and irrelevant IUs, such as “*in the*”, these word bigrams are always auxiliary phrases or parts of them and they do not carry much semantic information on discourse type. Also, the word bigrams related with the topic entity terms occur frequently in both relevant and irrelevant IUs, such as “*health care*”, “*star war*”. These word bigrams usually have nothing to do with the discourse type. Obviously, we cannot expect these word bigrams to occur frequently for different topics belonging to the same discourse types. However, some word bigrams occur more frequently in relevant IUs than irrelevant IUs, such as “*more than*”. If these word bigrams occur like

this for several topics belonging to the same discourse type, it's very likely that they are good indicators of a special discourse type.

Let's assume that the relevant set of a topic presented by query q consists of R documents: $rd_1, rd_2 \dots rd_R$ and the irrelevant set of a topic consists of I documents: $id_1, id_2 \dots id_I$. Let RIU_{ij} be the j -th IU in relevant document rd_i and $I IU_{ij}$ be the j -th IU in irrelevant document id_i .

Relevant Document Frequency (RDF)

Given a word sequence and some IUs of different relevant documents, the number of relevant documents that contain the word sequence is called relevant document frequency, RDF in short. If any IU of a relevant document contains the word sequence, we say that the document contains the word sequence. We use $RDF(q, ws)$ to denote RDF of word sequence ws in query q , which is the number of relevant documents in the retrieved list of query q that contain ws in their IUs.

The retrieved lists of different topics contain the different numbers of relevant documents. So in order to make comparison of the document frequencies of a word sequence among different topics, we need to normalize relevant document frequency. A simple way is to normalize RDF is to consider the number of relevant documents by:

$$\frac{RDF(q, ws)}{R(q)}$$

where $R(q)$ is the number of relevant documents for the topic present by query q . The results of above formula can also be regarded as the percentage of relevant documents that contain the word sequence.

We also propose “relevant document frequency N” as an extended feature to relevant document frequency. Relevant document frequency N, denoted by $RDF-n$, is the number of the relevant documents that contain a word sequence for no less than n times in the IUs of these relevant documents. In other words, for a given word sequence, $RDF-n$ ignores the relevant documents which contain the word sequence for less than n times when counting relevant documents. Obviously, RDF is a special instance of $RDF-n$, in which n is equal to one. RDF is always no less than $RDF-n$ and $RDF-n$ becomes smaller with the increase of n since the document selection condition becomes stricter.

Relevant IU Frequency (RIUF)

A document may contribute more than one IU if it contains more than one topic entity term. Relevant IUs are the IUs extracted from the relevant documents in the retrieved list in respond to a topic. Among all the IUs of a document, some of the IUs contain a given word sequence and others do not. Relevant IU frequency, denoted by $RIUF$, is the number of IUs that contain a given word sequence among all IUs in our investigation. We use $RIUF(rd, ws)$ to denote document RIUF, which is the IU frequency of the word sequence ws in the IUs of relevant document rd . We use $RIUF(q, ws)$ to denote query RIUF, which indicates the total IU frequency of the word sequence ws in the IUs of all the relevant documents of query q .

The numbers of IUs that belong to different relevant documents are not same so RIUF also needs to be normalized in order to compare the distribution of a word sequence in different relevant documents. Let us assume that $|rd|_{IU}$ is the number of

IUs that belong to the relevant document rd , like the normalization of RDF , we can normalize $RIUF$ into $RIUF(rd, ws)/|rd|_{IU}$, which can be regarded as the percentage of IUs that contain the word sequence ws for the relevant document rd .

The query RIUF is derived based on the document RIUF. For a given topic presented by query q , let the number of relevant documents in the retrieved list of query q be $R(q)$. we can aggregate the document RIUFs of all the relevant documents in the retrieved list in respond to q for a given word sequence ws into the query RIUF, which measures overall distribution of the given word sequence:

$$RIUF(q, ws) = \underset{i}{agg}[RIUF(rd_i, ws)]$$

In above expression, agg is an aggregation function which can be substituted by summation, fuzzy-set AND or OR functions etc. A simple way to derive above formula is to use summation to substitute function agg and then the result is the total number of IUs that contains ws in all the relevant documents in the retrieved list of query q :

$$RIUF(q, ws) = \sum_i RIUF(rd_i, ws)$$

Query RIUF can also be derived based on the normalized document RIUF. In the following formula, each item is normalized document RIUF, which is the percentage of relevant IUs that belong to one relevant document and contain the given word sequence. $R(q)$ is the number of relevant documents for the query q . The second way to compute query RIUF is the micro average of percentages of all the IUs that belong to all the relevant documents of the query q and contain the given word sequence ws , denoted by $RIUF_{micro}(q, ws)$. The micro percentage average weights equally all the documents, regardless of how many IUs belong to it. So we have:

$$RIUF_{micro}(q, ws) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} [RIUF(rd_i, ws) / |rd_i|_{IU}]$$

The third way to compute query RIUF is the macro average of percentages of all the IUs that belong to all the relevant documents of the query q and contain the given word sequence ws , which weights equally all the IUs, regardless of which relevant documents they belong to. The query RIUF based on the macro percentage average is denoted by $RIUF_{macro}(q, ws)$ and we compute it by:

$$RIUF_{macro}(q, ws) = \frac{\sum_{i=1}^{R(q)} RIUF(rd_i, ws)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}}$$

In summary, given a word sequence and a relevant document we propose the concept of document RIUF and normalized document RIUF considering the number of IUs that belong to this document. We also propose three ways to compute the query RIUF, which measures the distribution of the word sequence in the IUs of the retrieved relevant documents of a query. The simple one is the summation of the document RIUFs, which is the total number of the IUs in the relevant documents that contain the word sequence. The other two are based on normalized document RIUFs, which are aggregated by the micro and macro average of the percentages of the IUs of the relevant documents that contain the word sequence.

Word Sequence Frequency (WSF)

Word sequence frequency, WSF in short, is the number of the occurrences of a word sequence in different types of language unit including an IU of a relevant document, a relevant document and all the relevant documents in the retrieved list of a topic. The

WSF of a word sequence in an IU is called IU WSF. The WSF of a word sequence in a document is called document WSF. The WSF of all the relevant documents in the retrieved list of a topic presented by a query is called query WSF. These WSF-related features are similar with the concept “term frequency” commonly used in traditional information retrieval methods in that they all assume that the quantity of occurrence of a language unit (a term, a phrase, etc.) reflects the significance of the unit on determining the content of the document.

We use $freq(RIU_{ij}, ws)$ to denote the IU WSF of the word sequence ws in a relevant IU RIU_{ij} , which is the j -th IU extracted from the i -th relevant document rd_i . The basic unit in our study is IU extracted from a document; therefore, document WSF is the aggregation of the IU WSFs of all the IUs that belong to this document. We use $freq(rd_i, ws)$ to denote the document WSF of the word sequence ws in relevant document rd_i , agg_I to denote an aggregating function. So we have

$$freq(rd_i, ws) = agg_I [freq(RIU_{ij}, ws)]$$

A simple derivation of above formula is to substitute function agg_I by summation:

$$freq(rd_i, ws) = \sum_j freq(RIU_{ij}, ws)$$

In above formula, the document WSF is just the total number of occurrence of this word sequence in all the IUs that belong to the document.

Obviously, the numbers of IUs in different documents are not same, which also leads to the problem of normalization for document WSF. The normalizing factor should be related with the number of word sequences that can be possibly extracted from all the IUs of a document. Let us assume that an IU consists of m words (including the punctuations), we can successively extract $m-1$ word bigrams from this

IU. Obviously the first word bigram extracted consists of the first and the following second word; the last word bigram extracted consists of the $(m-1)$ th and m -th word. In the same way, we can justify that the number of word sequences of other lengths that can be extracted from a document depends on the number of IUs in the document. All IUs have the fixed size so the number of IUs in a document can be used as a normalizing factor. A normalized document WSF can be obtained by

$$norm_freq(rd_i, ws) = \frac{agg_1[freq(RIU_{ij}, ws)]}{|rd_i|_{IU}},$$

where $|rd_i|_{IU}$ is the number of IUs of the document rd_i . If the aggregating function in above formula is derived by summation, the normalized document WSF is

$$norm_freq(rd_i, ws) = \frac{\sum_j [freq(RIU_{ij}, ws)]}{|rd_i|_{IU}}$$

The result of above formula is the average IU WSF of the word sequence ws for all the IUs that belong to the document rd_i .

A topic (presented by a query in TREC) always has several relevant documents. We propose the query WSF to measure the distribution of word sequence in all relevant documents of a query. Let a topic be presented by query q . The query WSF of a word sequence is denoted by $Rfreq(q, ws)$. We compute the query WSF by the following formula where the aggregating function is denoted by agg_2 :

$$Rfreq(q, ws) = agg_2[\sum_i freq(rd_i, ws)] = agg_2\{agg_1[freq(RIU_{ij}, ws)]\}$$

A simple derivation of above formula is to use summation to substitute both aggregation functions:

$$Rfreq(q, ws) = \sum_i freq(rd_i, ws) = \sum_i \sum_j freq(RIU_{ij}, ws)$$

Above derivation is to compute the query WSF as the total number of occurrence of the word sequence ws in the IUs that belong to all the relevant documents of the query q .

There are two ways to compute the query WSF of a word sequence based on the normalized document WSF of the word sequence. We have derived the average word sequence frequency as the normalized word sequence of ws in the document rd_i by:

$$norm_freq(rd_i, ws) = \frac{\sum_j [freq(RIU_{ij}, ws)]}{|rd_i|_{IU}}$$

Hence, the normalized query WSF of the word sequence ws for query q can be obtained by the micro average of the document WSFs of ws for all the relevant documents of query q . The micro average weights equally the document WSFs of all the relevant documents, regardless how many IUs belong to it. Let $R(q)$ be the number of the relevant documents in the retrieved list of query q . We have:

$$\begin{aligned} & norm_Rfreq_{micro}(q, ws) \\ &= \frac{1}{R(q)} \sum_{i=1}^{R(q)} norm_freq(rd_i, ws) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} \frac{\sum_j [freq(RIU_{ij}, ws)]}{|rd_i|_{IU}} \end{aligned}$$

The second way of computing the normalized query WSF is to use the macro average of the document WSFs of the word sequence ws for all the relevant documents of query q . To compute the macro average, all the IU WSFs of the word sequence are summed up to obtain the document WSF, and then all the document WSFs are summed up to obtain the query WSF. In the same way, the total number of IUs of all the relevant documents is obtained and used as the denominator. We have:

$$\begin{aligned}
& norm_Rfreq_{macro}(q, ws) \\
&= \frac{\sum_{i=1}^{R(q)} freq(rd_i, ws)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} = \frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{|rd_i|_{IU}} freq(RIU_{ij}, ws)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}}
\end{aligned}$$

In summary, in the subsection, we put forward the concepts of IU WSF, document WSF and query WSF. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document WSF of a word sequence in a document. One is to sum up all the IU WSFs of the word sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU WSFs of all the IUs that belongs to the document. Second, we propose three ways to compute the query WSF of a word sequence in all the relevant documents of the retrieved list of the query q . The simple way is to use the total number of the occurrence of the word sequence in all the IUs of all the relevant documents, which is not normalized. The other ways are normalized by considering the number of IUs that belong to these relevant documents. They are respectively the micro and macro average of the IU WSFs of the word sequence in the IUs that belongs to all the relevant documents.

4.2.2 The distribution of word sequences in irrelevant documents

We can generalize rules or learn useful information from the relevant IUs. In addition, to use IUs extracted from irrelevant documents as negative instances will greatly enhance the discourse type machine learning process. As usual, a retrieved list that

responds to a TREC topic (query) contains 1000 documents. In this retrieved document list, the number of the irrelevant documents is always much bigger than that of relevant documents. So more negative instances are available for each topic if we assume the numbers of IUs in different documents are comparable.

Good instances always help learning a lot and bad instances will bring more noise. Hence, when we have abundant instances of different qualities, we need to select instances based on some criteria so that more valuable information is able to be learned from them. Compared with the limited number of positive instances, it's more necessary to create the criteria for instance selection for negative instances.

By carefully studying the content of some retrieved documents, we find that not all irrelevant documents are irrelevant only because they do not contain the information on the required discourse type. Some irrelevant documents are not relevant because they do not contain the information on the required entity at all, especially for the queries that need very specific information. For instance, an irrelevant document of the topic "advantage of Britain health care system" does contain the expression of "advantage" but it may talk about health care system of the United States. In this case, we cannot expect the IUs of this irrelevant document to provide us the negative information on the discourse type "advantage". Therefore, the IUs of the irrelevant documents that contain topic entity terms but do not contain the content on required discourse type should be used in learning as negative instances. These relevant documents always rank at the top of the retrieved list because they contain more topic entity terms in the IUs than other documents.

Irrelevant Document Selection

The criteria to select negative IUs are various. Theoretically, any criterion that is able to indicate the relevance of an IU to the topic entity is acceptable. A simple way is to use document similarity score obtained by traditional information retrieved model (e.g. BM11). This similarity score indicates to what extent a retrieved document is relevant to the topic entity. We believe that the irrelevant documents with higher similarity scores can provide better negatives instances than the ones with low similarity scores. However, in this way, all the IUs extracted from the same document have the same score. In other words, it's reasonable to use IUs of top ranked irrelevant documents in the retrieved list as a source of negative instances because we have more confidence that they contain information on topic entity and they are irrelevant because they do not contain the information on required discourse type.

Obviously, the above simple criterion is to assign the same score to all the IUs of the same irrelevant document, which is not accurate enough to indicate the difference among the IUs in a document since different IUs must be differently relevant with the topic. We know that the centre term of an IU is one of the topic entity terms. So it's likely that an IU is related with the topic entity but the relevance is small. We know that if a document is irrelevant, all the IUs from this document are all irrelevant. So for the IU extracted from an irrelevant document that is highly relevant to the topic entity, it's more likely that it does not contain the information on the required discourse type. Such an IU is a good negative instance for our learning. So the topic entity score of an IU is a qualified criterion for us to select the irrelevant IUs as negative instances.

We propose many ways of computing topic entity score for an IU in our fuzzy

model in Chapter 3, such as constant weighting measures, term feature based weighting measures and fuzzy set based weighting measures. For example, **DTF**, one of the constant weighting methods, can be a criterion for us to select irrelevant IUs. If we intend to use the IUs which contain all the topic entity terms as negative instances, we can set the criterion that the number of distinct entity terms should be equal to the number of the topic entity terms in the topic title. Another example **idf**, one of the term feature based weighting methods, can make us to evaluate the relevance of IUs by traditional tf-idf measurement.

Irrelevant Document Frequency (IDF)

Given a word sequence and some IUs of the selected different irrelevant documents, the number of irrelevant documents that contain the word sequence is irrelevant document frequency, *IDF* in short. If any of the IUs of an irrelevant document contains the word sequence, we say that the document contains the word sequence. We use $IDF(q, ws)$ to denote the number of irrelevant documents in the retrieved list of query q that contain word sequence ws in their IUs.

We know that the retrieved lists of different queries contain the different numbers of irrelevant documents. So in order to make comparison of the document frequencies of a word sequence among different topics, we need to normalize irrelevant document frequency. In the same with as what we do on *RDF*, A simple way to normalize *IDF* by considering the number of irrelevant documents is:

$$\frac{IDF(q, ws)}{I(q)}$$

where $I(q)$ is the number of selected irrelevant documents for a topic presented by

query q . The results of above formula can also be regarded as the percentage of relevant documents that contain the word sequence.

We also propose “irrelevant document frequency N ” as an extended feature to relevant document frequency. Irrelevant document frequency N , denoted by $IDF-n$, is the number of the irrelevant documents that contain a word sequence for no less than n times in the IUs of these irrelevant documents. We know that IDF is a special instance of $IDF-n$, in which n is equal to one. IDF is always no less than $IDF-n$ and $IDF-n$ becomes smaller with the increase of n since the document selection condition becomes stricter.

Irrelevant IU Frequency (IIUF)

An irrelevant document may contribute more than one IU if it contains more than one topic entity terms. Irrelevant IUs are the IUs extracted from the selected irrelevant documents of a topic. Some of the IUs of an irrelevant document contain a given word sequence and others do not. Irrelevant IU frequency, denoted by $IIUF$, is the number of IUs that contain a given word sequence among all IUs extracted from a irrelevant document. We use $IIUF(rd, ws)$ to denote the IU frequency of the word sequence ws in irrelevant document id . The numbers of IUs extracted from different irrelevant documents are not same so $IIUF$ also needs to be normalized in order to compare the distribution of a word sequence in different irrelevant documents. Let $|id|_W$ be the number of IUs extracted from relevant document id , like the normalization of $RIUF$, we normalize $IIUF$ into $IIUF(id, ws)/|id|_W$, which can be regarded as the percentage of IUs that contain the word sequence ws in irrelevant document id .

For a given topic presented by query q , we can aggregate the document IIUF of the IUs of all the selected irrelevant documents in the retrieved list in response to q for a given word sequence ws to measure the overall distribution of the given word sequence:

$$IIUF(q, ws) = \underset{i}{agg}[IIUF(id_i, ws)]$$

A simple way to derive above formula is to use summation to substitute function agg and then the result is the total number of IUs that contains ws for all the selected irrelevant documents in the retrieved list:

$$IIUF(q, ws) = \sum_i IIUF(id_i, ws)$$

In the following formula, each item is the percentage of irrelevant IUs of an irrelevant document that contain the given word sequence. $I(q)$ is the number of the selected irrelevant documents for the query q . We put forward the micro average of the percentages of all the irrelevant IUs that contain the given word sequence ws , denoted by $IIUF_{micro}(q, ws)$, which weights equally all the documents, regardless of how many IUs belong to it. So we have:

$$IIUF_{micro}(q, ws) = \frac{1}{I(q)} \sum_{i=1}^{I(q)} [IIUF(id_i, ws) / |id_i|_{IU}]$$

The following formula is macro average of the percentages of all the irrelevant IUs that contains the word sequence for the query q , which weights equally all the IUs.

The micro IIUF average is denoted by $IIUF_{macro}(q, ws)$ and we compute it by:

$$IIUF_{macro}(q, ws) = \frac{\sum_{i=1}^{I(q)} IIUF(id_i, ws)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}$$

In summary, given a word sequence and a query, we propose three ways to measure the distribution of the word sequence in the IUs of the selected retrieved irrelevant documents in response to the query. The simple one is the summation of the IIUF, which is the total number of the IUs in the selected irrelevant documents that contain the word sequence. The other two are normalized measures, which are the micro and macro average of the percentages of the IUs of the irrelevant documents that contain the word sequence.

Word Sequence Frequency (WSF)

The occurrence frequency-related measures of the distribution of word sequence in the selected irrelevant documents include IU WSF, document WSF and query WSF. We use $freq(IU_{ij}, ws)$ to denote the IU WSF of word sequence ws in an irrelevant IU IU_{ij} , which is the j -th IU extracted from the i -th irrelevant document id_i . Document WSF of a word sequence in an irrelevant document is the aggregation of the IU WSFs of all the IUs that belong to this document. We use $freq(id_i, ws)$ to denote the document WSF of the word sequence ws in irrelevant document id_i , and agg_I denotes an aggregating function. So we have

$$freq(id_i, ws) = agg_I [freq(IU_{ij}, ws)]$$

A simple derivation of above formula is to substitute function agg_I by summation:

$$freq(id_i, ws) = \sum_j freq(IU_{ij}, ws)$$

In above formula, the document WSF is just the total number of occurrence of this word sequence in all the IUs that belong to this irrelevant document.

Obviously, the numbers of IUs in different irrelevant documents are not same

either, which also leads to the problem of normalization. The normalizing factor is also related with the number of word sequences that can be possibly extracted from all the IUs of the irrelevant document. In the same way, the number of IUs in a document is used as a normalizing factor. A normalized document WSF can be obtained by

$$norm_freq(id_i, ws) = \frac{agg_1[freq(IU_{ij}, ws)]}{|id_i|_{IU}},$$

where $|id_i|_{IU}$ is the number of IUs of the document id_i . If the aggregating function in above formula is derived by summation, the normalized document WSF of the word sequence ws is

$$norm_freq(id_i, ws) = \frac{\sum_j [freq(IU_{ij}, ws)]}{|id_i|_{IU}}$$

The result of above formula is the average IU WSF of the word sequence ws for all the IUs that belong to the document id_i .

We also propose the query WSF to measure the distribution of word sequence in all the selected irrelevant documents of a topic. Let a topic be presented by query q . The query WSF of a word sequence is denoted by $Ifreq(q, ws)$. We compute the query WSF by the following formula where the aggregating function is denoted by agg_2 :

$$Ifreq(q, ws) = agg_2[freq(id_i, ws)] = agg_2 \{agg_1[freq(IU_{ij}, ws)]\}$$

A simple derivation of above formula is to use summation to substitute both aggregating functions:

$$Ifreq(q, ws) = \sum_i freq(id_i, ws) = \sum_i \sum_j freq(IU_{ij}, ws)$$

Above derivation is to compute the query WSF by the total number of occurrence of the word sequence ws in the IUs that belong to all the selected irrelevant documents of

the query q .

There are two ways to compute the query WSF of a word sequence based on the normalized document WSF of the word sequence. We have derived the average word sequence frequency as the normalized word sequence of ws in the document id_i by:

$$norm_freq(id_i, ws) = \frac{\sum_j [freq(IU_{ij}, ws)]}{|id_i|_W}$$

Hence, the normalized query WSF of the word sequence ws for query q can be obtained by the micro average of the document WSFs of ws for all the selected irrelevant documents of query q . The micro average weights equally the document WSFs of all the selected irrelevant documents, regardless how many IUs belong to it. Let $I(q)$ be the number of the selected irrelevant documents from the retrieved list of query q . We have:

$$\begin{aligned} norm_Ifreq_{micro}(q, ws) \\ = \frac{1}{I(q)} \sum_{i=1}^{I(q)} norm_freq(id_i, ws) &= \frac{1}{I(q)} \sum_{i=1}^{I(q)} \frac{\sum_j [freq(IU_{ij}, ws)]}{|id_i|_W} \end{aligned}$$

The second way of computing the normalized query WSF is to use the macro average of the document WSFs of the word sequence ws for all the selected irrelevant documents of query q . We use the same way to compute the macro average as we compute the macro average for relevant documents. We have:

$$\begin{aligned} norm_Ifreq_{macro}(q, ws) \\ = \frac{\sum_{i=1}^{I(q)} freq(id_i, ws)}{\sum_{i=1}^{I(q)} |id_i|_W} &= \frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{|id_i|_W} freq(IU_{ij}, ws)}{\sum_{i=1}^{I(q)} |id_i|_W} \end{aligned}$$

In summary, we put forward the concepts of IU WSF, document WSF and query WSF to measure the distribution of a word sequence in the IUs of the selected

irrelevant documents. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document WSF of a word sequence in a document. One is to sum up all the IU WSFs of the word sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU WSFs of all the IUs that belongs to the document. Second, we propose three ways to compute the query WSF of a word sequence in all the irrelevant documents of the retrieved list of the query q . The simplest way is to use the total number of the occurrence of the word sequence in all the IUs of all the irrelevant documents, which is not normalized. The other ways are normalized by considering the number of IUs that belong to these irrelevant documents. They are respectively the micro and macro average of the IU WSFs of the word sequence in the IUs that belongs to all the irrelevant documents.

4.2.3 Distribution features of word sequences

In this subsection, we will investigate and answer the following questions: if the distribution of a word sequence in the retrieved documents of a query is known, how to evaluate the ability of this word sequence to improve the retrieval. Hence, we propose some features based on the distribution of word sequence and then we can perform some retrospective experiments to evaluate these features in term of their ability of providing good word sequences to improve the discourse type based retrieval. For example, if a word sequence occurs in one of relevant IUs of a query and does not occur in any irrelevant IU of it, it will be definitely enhance the performance of retrieval for this query with the appropriate re-ranking formula.

We propose some features to sort word sequences and these features can be used in the later learning procedure. These features are based on the measures on the distribution of word sequence introduced in relevant and irrelevant IUs proposed in Section 4.2.1 and Section 4.2.2.

There are several ways to combine the measures that depend on the relevant documents and irrelevant documents. We derive some features by combining the relevant and irrelevant document related measures by ratio, which has the similar ideas with the log-likelihood ratio of relevance in binary independent retrieved model [Robertson 86] and signal-to-noise ratio. We know that signal-to-noise ratio (SNR) is an electrical engineering concept, which is also used in other fields such as scientific measurements, biological cell signaling, defined as the ratio of a signal power to the noise power corrupting the signal. Signal-to-noise ratio compares the level of a desired signal to the level of background noise. The higher the ratio, the less obtrusive the background noise is. In our features, the signal is a measure of the distribution of a word sequence in the IUs of relevant documents. The noise is a measure on the distribution of a word sequence in the irrelevant documents. Hence the feature will have a bigger result if a word sequence generally occur with a high frequency or occur in more IUs of the relevant documents or/and occur with a low frequency or occur in less IUs of the irrelevant documents.

Document Frequency Ratio (DFR)

We assume there is a topic set S consisting of T topics that have the same discourse type and they are presented by queries q_1, q_2, \dots, q_T . We define the query document

frequency ratio (query DFR) of a word sequence for a query as the ratio of the relevant document frequency (RDF) of the word sequence for a query to the irrelevant document frequency (IDF) to the query, as shown in below formula. A small constant α is to avoid division by zero.

$$DFR(q_i, ws) = \frac{RDF(q_i, ws)}{IDF(q_i, ws) + \alpha}$$

The query set DFR measures the distribution of a word sequence in the IUs of the relevant and irrelevant documents of a set of queries. Based on query DFR, we can propose a query set feature by combining the query set DFRs of all the queries in the query set. It is to aggregate the query DFR of each query in S by an aggregation function $agg(.)$ as:

$$DFR(S, ws) = \underset{q_i \in S}{agg} DFR(q_i, ws)$$

Note that above formula is a general form to compute query set DFR. We will derive this formula by using the different functions to substitute function agg in next subsection.

An extension of DFR is $DFR-n$, in which we use $RDF-n$ and $IDF-n$ to replace RDF and IDF in the calculation of DFR . Hence we have the following formula to compute $DFR-n$, which is more general and flexible:

$$DFR-n(q_i, ws) = \frac{RDF-n(q_i, ws)}{IDF-n(q_i, ws) + \alpha}$$

Considering that the numbers of the relevant and irrelevant documents of different queries are different, it's necessary to normalize query DFR. The normalized query DFR, denoted by $norm_DFR$, makes use of the normalized RDF and IDF of a word sequence. The normalized RDF and IDF of a word sequence in a set of documents are

actually the percentage of the documents that contain the word sequence in the IUs that belong to them. So each normalized RDF and IDF lies in the range between 0 and 1. Let us assume that the number of relevant documents in the retrieved list of query q_i is $R(q_i)$ so the normalized RDF is:

$$\frac{RDF(q_i, ws)}{R(q_i)}$$

Let the number of the selected irrelevant documents for learning be $I(q_i)$. The normalized IDF is

$$\frac{IDF(q_i, ws)}{I(q_i)}$$

So the normalized query DFR of a word sequence in the IUs of some selected retrieved documents of a query is

$$norm_DFR(q_i, ws) = \frac{RDF(q_i, ws) / R(q_i)}{[IDF(q_i, ws) + \alpha] / I(q_i)}$$

It is apparent that for the query q_i , if we select the same number of irrelevant and relevant documents, we have $R(q_i) = I(q_i)$, then the normalized query DFR have the same value as query DFR without normalization.

We propose the concept of normalized query set DFR to measure the distribution of a word sequence in the IUs of the selected retrieved documents for a set of queries with the consideration of the numbers of relevant and irrelevant documents of each query. A way to compute the normalized query set DFR is to use the micro average of the normalized query DFRs of the word sequence in all queries in the query set as:

$$\begin{aligned} & norm_DFR_{micro}(S, ws) \\ &= \frac{1}{T} \sum_{i=1}^T norm_DFR(q_i, ws) = \frac{1}{T} \sum_{i=1}^T \frac{RDF(q_i, ws) / R(q_i)}{[IDF(q_i, ws) + \alpha] / I(q_i)} \end{aligned}$$

We can see that if the same number of relevant and irrelevant documents are

selected as for each of the queries for learning, we have $R(q_i) = I(q_i)$, then the result of above formula, normalized DFR of a word sequence for a set of queries, has the same result as DFR without normalization. This case is just like the normalization of a word sequence for a query.

Another way to compute the normalized query set DFR is to compute the macro average of the query DFRs of the word sequence in all the queries in the query set as:

$$norm_DFR_{macro}(S, ws) = \frac{\sum_{i=1}^T [RDF(q_i, ws) / R(q_i)]}{\sum_{i=1}^T [IDF(q_i, ws) / I(q_i)]}$$

We find that when the same number of relevant and irrelevant documents are selected, viz. $R(q_i) = I(q_i)$, the normalized query set DFR is still different from the normalized query set DFR.

IU Frequency Ratio (IUFR)

In subsection 4.2.1 and 4.2.2 we know that given a word sequence and a query, there are three ways to compute the query RIUF and query IIUF of the word sequence based on the IU frequencies. The simple one is the summation of all the document IIUFs, which is the total number of the IUs in the selected irrelevant documents that contain the word sequence. The other two are normalized measures, which are the micro and macro average of the percentages of the IUs of the irrelevant documents that contain the word sequence.

IU Frequency ratio features are based on the above query RIUFs and query IIUFs, which can indicate the difference in the distribution of a word sequence in the IUs of

the relevant and irrelevant documents of a query. We use $IUFR(q_i, ws)$ to denote the query IUFR of the word sequence ws for the selected retrieved documents of query q .

A simple way to compute query IUFR is based on the query IIUF and query RIUF that are obtained by summing up all the document IIUFs and document RIUFs. In this function, the query IUFR is just the ratio of the total number of the IUs of the relevant documents to the one of the IUs of the selected irrelevant document. This method does not consider the difference in the numbers of IUs in the documents since the basic items are document RIUFs and document IIRFs without being normalized. Also, the method does not consider the possible difference in the number of selected relevant and irrelevance documents for learning. The function to compute query IUFR is as follows:

$$IUFR(q_i, ws) = \frac{RIUF(q_i, ws)}{IIUF(q_i, ws)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, ws)}{\sum_{i=1}^{I(q_i)} IIUF(id_i, ws)}$$

Query IUFR based on the normalized query RIUF and IIUF is shown by the below formula. The query RIUF and IIUF respectively come from the micro average of normalized document RIUFs and document IIUFs of the selected relevant and irrelevant documents of the query q_i . If the same number of relevant documents and irrelevant documents are selected for learning, the $R(q_i)$ and $I(q_i)$ can be cancelled. This method considers the number of IUs in different relevant and irrelevant documents. So it's a measure based on the normalized measures.

$$\begin{aligned}
& IUF R_{micro}(q_i, ws) \\
&= \frac{RIUF_{micro}(q_i, ws)}{IIUF_{micro}(q_i, ws)} = \frac{\frac{1}{R(q_i)} \sum_{i=1}^{R(q_i)} [RIUF(rd_i, ws) / |rd_i|_{IU}]}{\frac{1}{I(q_i)} \sum_{i=1}^{I(q_i)} [IIUF(id_i, ws) / |id_i|_{IU}]}
\end{aligned}$$

Furthermore, query IUF R can use normalized query RIUF and IIUF that come from the macro average of normalized document RIUFs and document IIUFs of the selected relevant and irrelevant documents of the query q_i , shown in the following formula. The macro average does not consider the number of the selected relevant and irrelevant documents of the query. So the calculation of query IUF R in this way does not consider the difference in the numbers of the selected relevant and irrelevant document either.

$$\begin{aligned}
& IUF R_{macro}(q_i, ws) \\
&= \frac{RIUF_{macro}(q_i, ws)}{IIUF_{macro}(q_i, ws)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, ws) / \sum_{i=1}^{R(q_i)} |rd_i|_{IU}}{\sum_{i=1}^{I(q_i)} IIUF(id_i, ws) / \sum_{i=1}^{I(q_i)} |id_i|_{IU}}
\end{aligned}$$

Based on the above features of query IUF R, we will discuss the calculation of query set IUF R, which indicates the distribution of a word sequence in the retrieved documents of a set of queries that have the same discourse type. We can think of the query set IUF R as the integrated aggregation of the different query IUF Rs of the queries in this set. In the following formula, $IUF R(S, ws)$ is the query set IUF R of the word sequence ws for the query set S , and the $agg(.)$ is an aggregating function.

$$IUF R(S, ws) = \underset{q_i \in S}{agg} IUF R(q_i, ws)$$

We will derive above formula to compute query set IUF R by using different aggregation functions in section 4.2.4.

Word Sequence Frequency Ratio (WSFR)

In the section 4.2.1 and 4.2.2, we put forward the concepts of IU WSF, document WSF and query WSF. We also proposed the calculation and normalization methods for these measures. There are three ways to compute the query WSF of a word sequence in all the relevant and selected irrelevant documents of the retrieved list of the query q . The first way is to use the total number of the occurrence of the word sequence in all the IUs of all the relevant documents, which is not normalized. The other two ways are normalized by considering the number of IUs that belong to these relevant documents. They are respectively the micro and macro average of the IU WSFs of the word sequence in the IUs that belongs to all the relevant documents.

Query word sequence frequency ratio (query WSFR) of a word sequence is based on its query WSFs in the retrieved documents of the query. For query q and word sequence ws , we define the query word sequence frequency ratio (query WSFR) as the ratio of the query word sequence frequency (query WSF) of ws in the relevant documents to the query WSF of ws in the selected irrelevant documents. Let $WSFR(q_i, ws)$ be the query WSF of q in the retrieved documents of query q and we define:

$$WSFR(q_i, ws) = \frac{Rfreq(q_i, ws)}{Ifreq(q_i, ws)}$$

If we use the query WSF of ws based on summation in the calculation of query WSFR, we have

$$WSFR(q_i, ws) = \frac{Rfreq(q_i, ws)}{Ifreq(q_i, ws)} = \frac{\sum_i \sum_j freq(RIU_{ij}, ws)}{\sum_i \sum_j freq(IIU_{ij}, ws)}$$

In above formula, the query WSFR of q_i is finally derived into the ratio of the

summation of the IU word sequence frequencies. This is a simple way but not normalized since the number of the IUs in the retrieved documents are is not considered.

We can also derive the query WSFR by using the normalized query WSFs based on micro and macro averages measures. The following two formulas show the calculation of the normalized query WSF.

$$WSFR_{micro}(q_i, ws) = \frac{norm_Rfreq_{micro}(q_i, ws)}{Inorm_Ifreq_{micro}(q_i, ws)} = \frac{\frac{1}{R(q)} \sum_{i=1}^{R(q)} \sum_j [freq(RIU_{ij}, ws)]}{\frac{1}{I(q)} \sum_{i=1}^{I(q)} \sum_j [freq(IIU_{ij}, ws)]}$$

$$WSFR_{macro}(q_i, ws) = \frac{norm_Rfreq_{macro}(q_i, ws)}{Inorm_Ifreq_{macro}(q_i, ws)} = \frac{\frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{|rd_i|_{IU}} freq(RIU_{ij}, ws)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}}}{\frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{|id_i|_{IU}} freq(IIU_{ij}, ws)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}}$$

In summary, we propose some features based on the measures of word sequence's distribution which are introduced in last two sections. These features are DFR which is based on the document frequency of a word sequence, IUFR which is based on the IU frequency of a word sequence and WSFR which is based on the word sequence frequency. These features indicate the distribution of a word sequence in the IUs of the selected relevant and irrelevant documents of a query so they are query-level features. In next subsection, these query-level features will be derived into query set-level features which indicate the distribution of a word sequence in all the selected retrieved

documents of all the queries with the same discourse type.

4.2.4 Cross validation experiments based on word sequences

In this subsection, we report our experimental results by using a method that is similar with K -fold cross validation to evaluate our word sequence discourse type model.

Cross-validation [Devijver 82], which is also called rotation estimation, is to partition a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subset(s) are retained for subsequent use in confirming and validating the initial analysis. The initial subset of data is called the training set; the other subset(s) are called validation or testing sets.

In K -fold cross-validation, the original sample is partitioned into K subsets. Among the K sample subsets, a single subset is kept as the validation data for testing the model, and the remaining $K-1$ subsets are used as training data. The cross-validation process is then repeated for K times, with each of the K subsets used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. In our experiments, K is equal to the number of queries that belong to the query set with the same discourse types. When one query is used for testing, all the relevant and irrelevant documents of the others queries can be used as training data.

Let a query set S contain n queries q_1, q_2, \dots, q_n with the same discourse type. When we use q_1 for testing, we obtain the retrieved lists of all the other queries and extract all the relevant documents of each query. Then the same number of irrelevant documents are extracted from the top of each retrieved list. We have explained that it's more likely for a top ranked irrelevant document to be related with topic entity (so it can be retrieved and ranked top by traditional retrieval methods) but it does not contain the information on discourse types. Therefore, top ranked irrelevant documents are better negative instances than the ones whose ranking positions are low. Then we extracted IUs from all the selected relevant and irrelevant documents based on the topic entity terms of their corresponding query. All the word sequences of the IUs of the relevant documents are extracted to compose a set A . Every element a_i of set A have a feature set $\{m_j(a_i)\}$ which are composed of the query set features of a_i in the IUs extracted from the selected retrieved documents of q_2, \dots, q_n . A feature evaluation function $F(.)$ is to evaluation the ability of a_i to improve the retrieval of q_2, \dots, q_n based on the feature set of a_i . So we use $F(\{m_j(a_i)\})$ to predict the ability of a_i to improve the retrieval of query q_1 . Therefore, in word sequence discourse type model we derived formula 4.2.2 into:

$$H(IU_i, dt) = f_{dt}[WS(IU_i)] = \underset{ws \in WS(IU_i)}{agg} [F(\{m_j(ws)\})]$$

which shows that we aggregate the function values (by $F(.)$) of each word sequence occurring in the i -th IU by aggregation function agg .

In order to make a very direct comparison among different features, we propose the function $F(.)$ by the following steps:

- (1) According to the given distribution feature m_j , we obtain the values of this

feature of all the elements in set A .

- (2) We sort all the elements in set A according to the values of their m_j feature.

Then we build a subset A_N of A with the top N elements.

- (3) For each element in set A , $F(m_j)$ returns a value based on whether this element belongs to A_N :

$$F(m_j(ws)) = \begin{cases} 1, & ws \in A_N \\ 0, & ws \notin A_N \end{cases}$$

Given a constant N and a feature m_j , we further derive formula 4.2.2 by counting how many word sequences occurring IU that also occur in A_N . Experiments shows that it's a simple and effective way. Also, we can change the value of N to make comparison among the measures and evaluation functions. Then the re-ranking formula 4.2.1 can be derived into:

$$\begin{aligned} rel(doc, q) &= \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))] \\ &= \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} + \sum_{ws \in WS(IU_i)} F(m_j(ws)) \right\} \end{aligned}$$

In this formula, $rel(doc, q)$ is the new score for re-ranking a document. S_0 is the original similarity score obtained by our baseline retrieval model and the original retrieved list is sorted according to S_0 . $|doc|_{IU}$ is the number of IUs in the document doc which is equal to the number of the topic entity terms in doc . $IU_i(doc)$ is the i -th IU of document doc . $tf(ws, WS(IU_i))$ is the total number of the word sequence ws occurring in the i -th IU. We use summation to combine this number with the previous item as what we did in retrospective experiments. We assign the original score $S_0/|doc|_{IU}$ to each IU in a document in order to quantify the contribution of an IU to the whole document, since the number of IUs in the documents are quite different.

After we use this formula to re-ranking the retrieved list of query q_1 , we obtain a new MAP of the re-ranked retrieved list. In the same way, we can have a new MAP for each of the other $n-1$ queries. Finally, we compute the mean of the MAPs of all the n queries as the results of the query set with this discourse type.

Table 4.2.2 Cross validation re-ranking retrieval performance based on word bigrams and feature DF m2

Word Sequence Sorting Feature: DF m2								
Disc Type	baseline	Top N of discourse type related word bigrams, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.0688	.0870	.0990	.1504	.1615	.2070	.2103
Reason	.2241	.1900	.1944	.2199	.2361	.2411	.2422	.2400
Impact	.2291	.1888	.2019	.2109	.2264	.2413	.2487	.2599 [#]
Mean	.2219	.1622	.1710	.1839	.2041	.2105	.2250	.2258

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

Table 4.2.3 Cross validation re-ranking retrieval performance based on word bigrams and feature QF n2

Word Sequence Sorting Feature: QF n2								
Disc Type	baseline	Top N of discourse type related word bigrams, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.2236	.2188	.2220	.2280	.2330	.2344	.2361
Reason	.2241	.2431	.2501	.2490	.2490	.2487	.2487	.2490 [^]
Impact	.2291	.2336	.2394	.2435	.2458	.2503	.2540	.2556
Mean	.2219	.2338	.2370	.2393	.2419	.2450	.2469	.2482

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

We present the results of the cross-validation experiments based on word bigrams with feature DF m2 and QF n2 respectively in Table 4.2.2 and 4.2.3. Although the

mean MAPs of all the discourse types are improved, only two runs are significantly improved, which are marked by the special symbol in both tables. The two runs belong to two different discourse types hence the results lack of consistency. Most of the runs are not significantly improved, which shows that word bigrams have very limited capability to improve all the discourse types by re-ranking.

In conclusion, the retrospective experiments show that word bigrams perform better than word trigrams and word 4-grams. We also concludes that the longer word sequence cannot offer better results than word bigrams with the analysis on the distribution of word sequences with different lengths. Among all the features, DFR m2 (micro average of the query DFR) and QF n2 (ratio of the sum of query RDFs to the sum of query IDFs) are two best features. Cross-validation experiments show that word bigrams have ability to improve some of the queries of all the three discourse types but they cannot consistently improve all the queries to a statistically significant level.

4.3 Discourse Type Based Retrieval by using POS Tag Sequences

In the section, we will investigate and evaluate POS tag sequence features based on POS tag sequence discourse type model. Rather than study every single POS tags, we investigate the POS tag sequences that consist of at least two tags. A POS tag sequence is called POS tag bigram, POS trigram and POS tag N-gram if its length is two, three and N respectively.

In Table 4.2.1 of section 4.2, we show an example of an IU and the word bigrams

and trigrams that are possibly extracted from it. We also use the same IU as an example to show the POS tag bigrams and trigrams. Table 4.3.1 illustrates the same IU that is extracted from document whose DOCID is LA031589-0075, which is one the retrieved documents of query No.690. We tag the IU POS tags with the POS tagger Monty Tagger [Liu 04] which is a rule-based part-of-speech tagger based on Eric Brill's transformational-based learning POS tagger [Brill 94], and uses Brill-compatible lexicon and rule files. Monty Tagger uses the University of Pennsylvania (Penn) Treebank Tag-set [Santorini 90]. During the POS tag sequence extraction, POS tag sequences with different lengths can be extracted sequentially from the first tag. The first POS tag sequence begins with the first tag and the last one ends with the last tag. For the reason brevity, we just show some of the POS tag bigrams and trigrams in the table.

Table 4.3.1 An example of an IU and the extracted POS tag bigrams and trigram

690 LA031589-0075	
the/DT promise/NN of/IN economic/JJ growth/NN and/CC jobs/NNS ,/, but/CC also/RB the/DT chance/NN to/TO develop/VB a/DT unique/JJ society/NN of/IN cultural/JJ ,/, <<educational/JJ >> ,/, technological/JJ and/CC artistic/JJ diversity/NN ./.. To/TO do/VB so/RB ,/, California/NNP must/MD invest/VB now/RB in/IN the/DT necessary/JJ public/JJ support/NN	
POS tag bigrams:	POS tag trigrams:
DT NN	DT NN IN
NN IN	NN IN JJ
IN JJ	IN JJ NN
JJ NN	JJ NN CC
NN CC	NN CC NNS
...	...
DT JJ	DT JJ JJ
JJ JJ	JJ JJ NN
JJ NN	

Apparently, the number of different POS tags (e.g. the number of tags in Penn's

Treebank tag-set) is much fewer than the number of distinct words. We can conclude that the number of POS tag sequences that are possibly extracted are also much fewer than POS tag sequences with the same length. Moreover, it's obvious that one POS tag sequence may correspond to a lot of different word sequences. These different word sequences may compose of different words however it's likely that they have very similar patterns or structures. For example, there are some common patterns or structures to express the positive/negative opinions, make comparison or analyze. Therefore, POS tag sequences can be used to detect and estimate the discourse type of the text.

Let us use POS tag sequence "VBZ RB JJ" as an example to show the ditribution of POS tag sequence and the mapping from a POS tag sequence to several word sequences. "VBZ RB JJ" corresponds to the word trigrams that sequentially compose of a VBZ word (verb, present tense, 3rd person singular), a RB word (adverb) and a JJ word (adjective or numeral, ordinal). We count the number of relevant documents and irrelevant documents (the same number as relevant documents and they are selectef from the top of the retrieved list) that contain "VBZ RB JJ" in their POS-tagged IUs. These numbers are shown in Table 4.3.2. For example, among the POS-tagged IUs of 48 relevant and 48 top-ranked irrelevant documents of query No.654, this POS tag trigram occurs in 20 of relevant documents and 7 irrelevant ones. From the statistics on the eight queries, we know it's helpful to improve most of the queries of the discourse type advantage/disadvantage.

Table 4.3.2 Query RDF, query IDF of POS tag trigram “VBZ RB JJ” for each topic

Topic ID	# of doc in rel set	# of rel doc containing it	# of irrel doc containing it	RDF>IDF?
308	4	1	0	Yes
605	63	28	7	Yes
608	25	6	2	Yes
624	18	3	1	Yes
637	22	5	3	Yes
654	48	20	7	Yes
690	6	1	3	No
699	66	10	5	Yes
Total	252	74	28	6

In order to see what word sequences respond to “VBZ RB JJ “, we check all the possible word trigrams in our study and find that the following ones shown in the Table 4.3.3. match the POS tag trigram. Please notice that we provide the words that just follow the matched word trigrams to provide more information so Table 4.3.3 actually contains word 4-grams. From these word 4-grams, we can see these expressions are generally related with the judgment or evaluations which are basic ways of stating advantages and disadvantages.

According to the Appraisal Theory [Martin 05], there are three linguistic ways to indicate an attitude: by reference to emotion (AFFECT), with respect to social norms (JUDGEMENT), by reference to aesthetic principles and other systems of social value (APPRECIATION). Then attitude forms evaluation which can be positive or negative. It is not hard to find that the expressions shown in Table 4.3.3 cover all the three ways. For instance, “*is particularly embarrassing*”, “*is usually zealous about*”, “*is so appealing that*” belong to AFFECT, “*s so out-dated*”, “*is very democratic*”, “*is especially anti-incumbent*” belong to “JUDGEMENT” and “*is theoretically possible*”,

“*is just flat-out wrongheaded*”, “*is certainly sufficient*” belong to “APPRECIATION”.

Table 4.3.3 Some examples of word trigrams with the same POS tags that may be related with advantage/disadvantage

VBZ RB JJ	
has enough conservative support	is so hard to
has only limited experience	is so little competition
is again considerable variation	is so low ,
is almost impossible to	is so obvious as
is already possible to	is so raw .
is also striking how	is so regressive --
is certainly sufficient .	is so tight .
is equally clear .	is solidly middle-class with
is equally fierce local	is still expensive .
is especially anti-incumbent ,	is still strong demand
is fully co-educational throughout	is theoretically possible ,
is hardly surprising that	is too much of
is heavily over-subscribed ,	is ultimately political .
is highly selective .	is usually zealous about
is increasingly true .	is very democratic ,
is just flat-out wrongheaded	is very different from
is not clear how	is very important that
is not clear whether	is very marginal .
is not evident that	is very much a
is not important .	is very much part
is not likely to	is very privileged .
is not responsive to	is virtually impossible to
is not surprising that	is virtually unthinkable .
is not true .	isn't just pie-in-the sky
is not valid .	looks very impressive .
is not worth the	looks very promising .
is now full of	's almost inevitable that
is now limited to	's not difficult to
is particularly embarrassing for	's not just politicians
is predominantly public or	's pretty clear that
is quite high .	's so out-dated ,
is quite possible that	's very doubtful that
is relatively cheap to	's very little variation
is relatively low .	seems quite realistic .
is similarly ambivalent about	signifies very little .
is so appealing that	

Note: the first three words of the word 4-grams have the same POS tag sequence and the fourth words (punctuations) are provided only in order to offer more linguistic and background information

POS tag sequence discourse type model is formulated to predict the probability that the discourse type of a part of text (IU, information unit) matches the desirable discourse type based on the POS tag sequences of different lengths (POS tag bigrams, trigrams, etc.) that occur in the POS tagged text. A POS tag sequence discourse type model can be formulated as:

$$rel(doc, q) = \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))] \quad (4.3.1)$$

In formula 4.3.1, $rel()$ denotes the relevance between a given document doc and query q , $sim()$ denotes the similarity between an IU and a query based on non-discourse type methods, such as vector space model or IU similarity model. $H()$ denotes the probability of IU has the discourse type dt based on the learned function. “ \wedge ” denotes a conjunctive function. A simple to way to derive $H()$ is as:

$$H(IU_i, dt(q)) = f_{dt}[PS(IU_i)] = \sum_{ps \in A} tf[ps, PS(IU_i)] \quad (4.3.2)$$

$wp(IU_i)$ is the data structure consisting of all the POS tag sequences with a certain length occurring in IU and their frequencies and positions. $f_{dt}[]$ is a function obtained from machine learning methods, which determines the probability of the discourse type of an IU, which is presented by a POS tag data structure $PS(IU_i)$, meets the desirable discourse type $dt(q)$ of the query q . One way of learning the function $f_{dt}[]$ is to learn a set A containing the POS tag sequences that are most likely to express the discourse type $dt(q)$. Then all POS tag sequences in an IU can be weighted according to whether it belongs to A , or, more complicated, its position in A if all elements are sorted in A . Therefore, we finally derive formula 4.3.2 into the number of POS tag sequences in tagged IU that occur in A , which is denoted by $tf[]$.

In next subsection, we will introduce the measures of the distribution of the POS tag sequences in the training documents. Some aggregation functions based on these measures provide different ways to derive formula 4.3.2.

4.3.1 The distribution of POS tag sequence in relevant documents

Let's assume that the relevant set of a topic presented by query q consists of R documents: $rd_1, rd_2 \dots rd_R$ and the irrelevant set of a topic consists of I documents: $id_1, id_2 \dots id_I$. Let RIU_{ij} be the j -th IU in relevant document rd_i and $I IU_{ij}$ be the j -th IU in irrelevant document id_i .

Relevant Document Frequency (RDF)

Given a POS tag sequence and some tagged IUs of different relevant documents, the number of relevant documents that contain the POS tag sequence is relevant document frequency, DRF in short. If any IU of a relevant document contains the POS tag sequence in their POS tags, we say that the document contains the POS tag sequence.

We use $RDF(q, ps)$ to denote the number of relevant documents in the retrieved list of query q that contain POS tag sequence ps in their IUs.

Relevant IU Frequency (RIUF)

Relevant IU frequency, denoted by $RIUF$, is the number of the POS-tagged IUs of the relevant documents that contain a given POS tag sequence. We use $RIUF(rd, ps)$ to denote document RIUF, which is the IU frequency of the POS tag sequence ps in the

IUs of relevant document rd . We use $RIUF(q, ps)$ to denote query RIUF, which indicates the total IU frequency of the POS tag sequence ps in the IUs of all the relevant documents of query q . Document RIUF can be normalized by $RIUF(rd, ws)/|rd|_{IU}$, where $|rd|_{IU}$ is the number of IUs that belong to the relevant document rd .

The query RIUF of a POS tag sequence is defined based on document RIUF. We can aggregate the document RIUFs of all the relevant documents in the retrieved list in response to q for a given POS tag sequence ps into the query RIUF, which measures overall distribution of the given POS tag sequence:

$$RIUF(q, ps) = \underset{i}{agg}[RIUF(rd_i, ps)]$$

A simple way to derive above formula is to use summation to substitute function agg and then the result is the total number of IUs that contains ps in all the relevant documents in the retrieved list of query q :

$$RIUF(q, ps) = \sum_i RIUF(rd_i, ps)$$

Query RIUF can also be computed based on the document RIUF by some more complicated ways, for example, the number of IUs in the relevant document can be considered. Let $R(q)$ be the number of relevant documents for the query q . The second way to compute query RIUF is the micro average of percentages of all the POS tagged IUs that belong to all the relevant documents of the query q and contain the given POS tag sequence ps . The micro percentage average weights equally all the documents, regardless of how many IUs belong to it. So we have:

$$RIUF_{micro}(q, ps) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} [RIUF(rd_i, ps) / |rd_i|_{IU}]$$

The third way to compute query RIUF is the macro average of percentages of all

the POS-tagged IUs that belong to all the relevant documents of the query q and contain the given POS tag sequence ps . We compute the query RIUF by using the macro percentage average as aggregation method as:

$$RIUF_{macro}(q, ps) = \frac{\sum_{i=1}^{R(q)} RIUF(rd_i, ps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}}$$

In summary, given a POS tag sequence and a relevant document we propose the concept of document RIUF. Moreover, we propose three ways to compute the query RIUF of a POS tag sequence. The simple one is the summation of the RIUF, which is the total number of the IUs in the relevant documents that contain the POS tag sequence. The other two are normalized measures, which are the micro and macro average of the percentages of the IUs of the relevant documents that contain the POS tag sequence.

POS tag sequence Frequency (PSF)

POS tag sequence frequency, PSF in short, is the number of the occurrence of a POS tag sequence in a POS tagged IU of a relevant document, a relevant document or all the relevant documents in the retrieved list of a topic. The PSF of a POS tag sequence in an IU is called IU PSF. The PSF of a POS tag sequence in a document is called document PSF. The PSF of all the relevant documents in the retrieved list of a topic presented by a query is called query PSF.

We use $freq(RIU_{ij}, ps)$ to denote the IU PSF of POS tag sequence ps in a relevant IU RIU_{ij} (the j -th IU extracted from the i -th relevant document rd_i). Document PSF is the aggregation of the IU PSFs of all the IUs that belong to this document. We

use $freq(rd_i, ps)$ to denote the document PSF of the POS tag sequence ps in relevant document rd_i , agg_1 to denote an aggregating function. So we have

$$freq(rd_i, ps) = agg_1 [freq(RIU_{ij}, ps)]$$

A simple derivation of above formula is to substitute function agg_1 by summation:

$$freq(rd_i, ps) = \sum_j freq(RIU_{ij}, ps)$$

In above formula, the document PSF is the total number of occurrence of this POS tag sequence in all the IUs that belong to the document.

There is also a problem of normalization for PSF measures. Obviously, we can justify that the number of POS tag sequences of other lengths that can be possibly extracted from a document depends on the size of IU. All IUs have the fixed size so the number of IUs in a document, instead of the length of document, can be used as a normalizing factor. A normalized document PSF can be obtained by

$$norm_freq(rd_i, ps) = \frac{agg_1 [freq(RIU_{ij}, ps)]}{|rd_i|_{IU}}$$

where $|rd_i|_{IU}$ is the number of IUs of the document rd_i . If the aggregating function in above formula is derived by summation, the normalized document PSF is

$$norm_freq(rd_i, ps) = \frac{\sum_j [freq(RIU_{ij}, ps)]}{|rd_i|_{IU}}$$

The result of above formula is the average IU PSF of the POS tag sequence ps for all the IUs that belong to the document rd_i .

Let a topic be presented by query q . The query PSF of a POS tag sequence is denoted by $Rfreq(q, ps)$. We compute the query PSF by the following formula where the aggregating function is denoted by agg_2 :

$$Rfreq(q, ps) = agg_2 [freq(rd_i, ps)] = agg_2 \{agg_1 [freq(RIU_{ij}, ps)]\}$$

A simple derivation of above formula is to use summation to substitute both aggregating functions:

$$Rfreq(q, ps) = \sum_i freq(rd_i, ps) = \sum_i \sum_j freq(RIU_{ij}, ps)$$

There are also two ways to compute the query PSF of a POS tag sequence based on the normalized document PSF of the POS tag sequence. The normalized query PSF of the POS tag sequence ps for query q can be obtained by the micro average of the document PSFs of ps for all the relevant documents of query q as:

$$\begin{aligned} & norm_Rfreq_{micro}(q, ps) \\ &= \frac{1}{R(q)} \sum_{i=1}^{R(q)} norm_freq(rd_i, ps) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} \frac{\sum_j [freq(RIU_{ij}, ps)]}{|rd_i|_{IU}} \end{aligned}$$

The second way of computing the normalized query PSF is to calculate the macro average of the document PSFs of the POS tag sequence ps for all the relevant documents of query q :

$$\begin{aligned} & norm_Rfreq_{macro}(q, ps) \\ &= \frac{\sum_{i=1}^{R(q)} freq(rd_i, ps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} = \frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{|rd_i|_{IU}} freq(RIU_{ij}, ps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} \end{aligned}$$

In summary, in the subsection, we put forward the concepts of IU PSF, document PSF and query PSF. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document PSF of a POS tag sequence in a document. One is to sum up all the IU PSFs of the POS tag sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU PSFs of all the IUs that belongs to the document. Second, we propose three ways to

compute the query PSF of a POS tag sequence in all the relevant documents of the retrieved list of the query q . They are summation, micro and macro average of the IU PSFs of the POS tag sequence in the IUs that belongs to all the relevant documents.

4.3.2 The distribution of POS tag sequence in irrelevant documents

In order to make comparison among the experimental results based on different discourse type models, we use the same way to select the irrelevant documents for POS tag sequence discourse type model as we do for word sequence discourse type model. For each query, we select the same number of the irrelevant documents from the beginning of the retrieved list according to the number of relevant documents of the query. We propose the following measures to reflect the distribution of POS tag sequence in the IUs of the selected irrelevant documents.

Irrelevant Document Frequency (IDF)

In the training set of the selected irrelevant documents, the number of the irrelevant documents that contain the POS tag sequence is called irrelevant document frequency, IRF. If any IU of an irrelevant document contains the POS tag sequence in their POS tags, we say that the document contains the POS tag sequence. We use $IDF(q, ps)$ to denote IDF of the POS tag sequence ps in the retrieved list of query q .

Irrelevant IU Frequency (IIUF)

Irrelevant IU frequency, denoted by $IIUF$, is the number of the POS-tagged IUs of the irrelevant documents that contain a given POS tag sequence. We use $IIUF(id, ps)$ to denote document IIUF, which is the IU frequency of the POS tag sequence ps in the IUs of irrelevant document id . We use $IIUF(q, ps)$ to denote query IIUF, which denotes the total IU frequency of the POS tag sequence ps in the IUs of all the irrelevant documents of query q . Document IIUF can be normalized by $IIUF(id, ps)/|id|_{IU}$, where $|id|_{IU}$ is the number of IUs that belong to the irrelevant document id .

The query IIUF of a POS tag sequence is defined based on document IIUF. We can aggregate the document RIUFs of all the selected irrelevant documents in the retrieved list in respond to q for a given POS tag sequence ps into the query RIUF. A simple way to derive above formula is to use summation:

$$IIUF(q, ps) = \sum_i IIUF(id_i, ps)$$

Query IIUF can also be computed by using the micro and macro average as aggregation function. Let $I(q)$ be the number of selected irrelevant documents for the query q . We have the following formula by using micro percentage average and macro percentage average:

$$IIUF_{micro}(q, ps) = \frac{1}{I(q)} \sum_{i=1}^{I(q)} [IIUF(id_i, ps) / |id_i|_{IU}]$$

$$IIUF_{macro}(q, ps) = \frac{\sum_{i=1}^{I(q)} IIUF(id_i, ps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}$$

In summary, given a POS tag sequence and an irrelevant document we propose the concept of document IIUF. Moreover, we propose three ways to compute the query RIUF of a POS tag sequence by using different aggregation functions: summation,

micro average and macro average.

POS tag sequence Frequency (PSF)

There are also three levels of PSF. The PSF (POS tag sequence frequency) of a POS tag sequence in an IU is called IU PSF. The PSF of a POS tag sequence in a document is called document PSF. The PSF of all the selected irrelevant documents in the retrieved list of a topic presented by a query is called query PSF.

We use $freq(IU_{ij}, ps)$ to denote the IU PSF of the POS tag sequence ps in irrelevant IU IU_{ij} . Document PSF is the aggregation of the IU PSFs of all the IUs that belong to this document. We use $freq(id_i, ps)$ to denote the document PSF of the POS tag sequence ps in irrelevant document id_i . A simple way to compute document PSF is to aggregate the IU PSFs by summation:

$$freq(id_i, ps) = \sum_j freq(IU_{ij}, ps)$$

In above formula, the document PSF is the total number of occurrence of this POS tag sequence in all the IUs that belong to the document.

There is also a problem of normalization for PSF measures. A normalized document PSF can be obtained by

$$norm_freq(id_i, ps) = \frac{agg_j[freq(IU_{ij}, ps)]}{|id_i|_{IU}},$$

where $|id_i|_{IU}$ is the number of IUs of the document id_i . If the aggregating function in above formula is derived by summation, the normalized document PSF is

$$norm_freq(id_i, ps) = \frac{\sum_j [freq(IU_{ij}, ps)]}{|id_i|_{IU}}$$

The result of above formula is the average IU PSF of the POS tag sequence ps for all

the IUs that belong to the document id_i .

Let a topic be presented by query q . The query PSF of a POS tag sequence is denoted by $Rfreq(q, ps)$. We compute the query PSF by the following formula based on the aggregating function agg_2 :

$$Rfreq(q, ps) = agg_2[freq(id_i, ps)] = agg_2 \{ agg_1[freq(IIU_{ij}, ps)] \}$$

A simple derivation of above formula is to use summation to substitute both aggregating functions:

$$Rfreq(q, ps) = \sum_i freq(id_i, ps) = \sum_i \sum_j freq(IIU_{ij}, ps)$$

There are also two ways to compute the query PSF of a POS tag sequence based on the normalized document PSF of the POS tag sequence. The normalized query PSF of the POS tag sequence ps for query q can be obtained by the micro average of the document PSFs of ps for all the selected irrelevant documents of query q as:

$$\begin{aligned} & norm_Rfreq_{micro}(q, ps) \\ &= \frac{1}{I(q)} \sum_{i=1}^{I(q)} norm_freq(id_i, ps) = \frac{1}{I(q)} \sum_{i=1}^{I(q)} \frac{\sum_j [freq(IIU_{ij}, ps)]}{|id_i|_{IU}} \end{aligned}$$

The second way of computing the normalized query PSF is to calculate the macro average of the document PSFs of the POS tag sequence ps for all the selected irrelevant documents of query q :

$$\begin{aligned} & norm_Rfreq_{macro}(q, ps) \\ &= \frac{\sum_{i=1}^{I(q)} freq(id_i, ps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}} = \frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{|id_i|_{IU}} freq(IIU_{ij}, ps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}} \end{aligned}$$

In summary, in this subsection, we put forward the concepts of IU PSF, document

PSF and query PSF to measure the distribution of a POS tag sequence in the IUs of irrelevant documents. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document PSF of a POS tag sequence in a document. One is to sum up all the IU PSFs of the POS tag sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU PSFs of all the IUs that belongs to the document. Second, we propose three ways to compute the query PSF of a POS tag sequence in all the selected irrelevant documents of the retrieved list of the query q . They are summation, micro and macro average of the IU PSFs of the POS tag sequence in the IUs that belongs to all the select irrelevant documents.

4.3.3 Distribution features of POS tag sequences

In this subsection, we propose some features to evaluate POS tag sequence and these features can be used in the later learning procedure. These features are based on the measures on the distribution of POS tag sequence introduced in previous subsections. We also use the ratio of the measure obtained from relevant set to the measure obtained from the corresponding irrelevant set.

Document Frequency Ratio (DFR)

We assume there is a topic set S consisting of T topics that have the same discourse type and they are presented by queries q_1, q_2, \dots, q_T . We define the query document

frequency ratio (query DFR) of a POS tag sequence for a query as the ratio of the relevant document frequency RDF of the POS tag sequence for a query to the irrelevant document frequency IDF to the query, as shown in below formula. A small constant α is to avoid division by zero.

$$DFR(q_i, ps) = \frac{RDF(q_i, ps)}{IDF(q_i, ps) + \alpha}$$

The query set DFR measures the distribution of a POS tag sequence in the IUs of the relevant and irrelevant documents of a set of queries. Based on query DFR, we can propose a function to aggregate the query DFRs of all the queries in the query set. It is to aggregate the query DFR of each query in S by an aggregation function $agg(.)$ as:

$$DFR(S, ps) = \underset{q_i \in S}{agg} DFR(q_i, ps)$$

We will further derive above formula in next subsection by substitute the function agg by different operations.

IU Frequency Ratio (IUFR)

IU Frequency ratios are measures based on query RIUF and query IIUF, which can indicate the distribution of a POS tag sequence in the IUs of the relevant and irrelevant documents of a query. We use $IUFR(q_i, ps)$ to denote the query IUFR of the POS tag sequence ps for the selected retrieved documents of query q .

The simplest way to compute query IUFR is based on the query IIUF and query RIUF that are obtained by summing up all the document IIUFs and document RIUFs. The feature is the ratio of two summations:

$$IUF R(q_i, ps) = \frac{RIUF(q_i, ps)}{IIUF(q_i, ps)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, ps)}{\sum_{i=1}^{I(q_i)} IIUF(id_i, ps)}$$

Query IUF R can use normalized query RIUF and IIUF. For example, the following formula uses query RIUF and query IIUF that come from the macro average of normalized document RIUFs and document IIUFs:

$$IUF R_{macro}(q_i, ps) = \frac{RIUF_{macro}(q_i, ps)}{IIUF_{macro}(q_i, ps)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, ps) / \sum_{i=1}^{R(q_i)} |rd_i|_{IU}}{\sum_{i=1}^{I(q_i)} IIUF(id_i, ps) / \sum_{i=1}^{I(q_i)} |id_i|_{IU}}$$

The calculation of query set IUF R is based on the above measures of query IUF R, which indicates the distribution of a POS tag sequence in the retrieved documents of a set of queries with the same discourse type. In the following formula, $IUF R(S, ps)$ is the query set IUF R of the POS tag sequence ps for the query set S , and the $agg(.)$ is an aggregating function.

$$IUF R(S, ps) = \underset{q_i \in S}{agg} IUF R(q_i, ps)$$

We will derive above formula in section 4.3.4 to compute query set IUF R by using different aggregation functions.

POS tag Sequence Frequency Ratio (PSFR)

In the subsection 4.3.1 and 4.3.2, we put forward the concepts of PSF in three levels: IU PSF, document PSF and query PSF. We also derived the calculation and normalization methods for these measures. In this subsection, we will propose the

concept of POS tag Sequence Frequency Ratio (PSFR) and the calculation of PSFR.

Query POS tag sequence frequency ratio (query PSFR) of a POS tag sequence is based on its query PSFs in the retrieved documents of the query. For query q and POS tag sequence ps , we define query PSFR as the ratio of the query POS tag sequence frequency (query PSF) of ps in the relevant documents to the query PSF of ps in the selected irrelevant documents. Let $PSFR(q_i, ps)$ be the query PSF of q in the retrieved documents of query q and according to our definition, we have:

$$PSFR(q_i, ps) = \frac{Rfreq(q_i, ps)}{Ifreq(q_i, ps)}$$

We can derive above formula by using the query PSF of ps based on summation in the calculation of query PSFR, we have

$$PSFR(q_i, ps) = \frac{Rfreq(q_i, ps)}{Ifreq(q_i, ps)} = \frac{\sum_i \sum_j freq(RIU_{ij}, ps)}{\sum_i \sum_j freq(IIU_{ij}, ps)}$$

In above formula, the query PSFR of q_i is finally derived into the ratio of the summation of the IU POS tag sequence frequencies. This is a simple way but not normalized since the number of the IUs in the retrieved documents are is not considered.

We can also derive the query PSFR by using the normalized query PSFs based on micro and macro averages measures. The following two formulas show the calculation of the normalized query PSF.

$$PSFR_{micro}(q_i, ps) = \frac{norm_Rfreq_{micro}(q_i, ps)}{Inorm_Ifreq_{micro}(q_i, ps)} = \frac{\frac{1}{R(q)} \sum_{i=1}^{R(q)} \frac{\sum_j [freq(RIU_{ij}, ps)]}{|rd_i|_{IU}}}{\frac{1}{I(q)} \sum_{i=1}^{I(q)} \frac{\sum_j [freq(IIU_{ij}, ps)]}{|id_i|_{IU}}}$$

$$PSFR_{macro}(q_i, ps) = \frac{norm_Rfreq_{macro}(q_i, ps)}{Inorm_Ifreq_{macro}(q_i, ps)} = \frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{rd_i|IU} freq(RIU_{ij}, ps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} \cdot \frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{id_i|IU} freq(IIU_{ij}, ps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}$$

In summary, we propose some features with the ratio form based on the measures of POS tag sequence's distribution which are introduced in last two sections, including DFR which is based on the document frequency of a POS tag sequence, IUFR which is based on the IU frequency of a POS tag sequence and PSFR which is based on the POS tag sequence frequency. These query-level features indicate the distribution of a POS tag sequence in the IUs of the selected relevant and irrelevant documents of a query. In next section, these query-level features will be derived into query set-level features which indicate the distribution of a POS tag sequence in all the selected retrieved documents of all the queries with the same discourse type.

4.3.4 Cross validation experiments based on POS tag sequence

In this subsection, we report our experimental results by using a method that is similar with K-fold cross validation to evaluate our POS-tag sequence discourse type model. The method is same with the method used for word discourse type model. So we don't redundantly introduce the method.

All the POS tag sequences of the tagged text of the IUs of the relevant documents are extracted to compose a set A . Every element a_i of set A have a feature set $\{m_j(a_i)\}$. A feature evaluation function $F(.)$ is to evaluation the ability of a_i to improve the retrieval of the training queries based on the feature set of a_i . So we use $F(\{m_j(a_i)\})$ to predict the ability of a_i to improve the retrieval of the testing query. Therefore, in POS tag sequence discourse type model we derived formula 4.3.2 into:

$$H(IU_i, dt) = f_{dt}[PS(IU_i)] = \underset{ps \in PS(IU_i)}{agg} [F(\{m_j(ps)\})]$$

which shows that we aggregate the function values (by $F(.)$) of each POS tag sequence occurring in the i -th IU by aggregation function agg .

In order to make a very direct comparison among different features, we propose the function $F(.)$ by the following steps:

- (4) According to the given distribution feature m_j , we obtain the values of this feature of all the elements in set A .
- (5) We sort all the elements in set A according to the values of their m_j feature. Then we build a subset A_N of A with the top N elements.
- (6) For each element in set A , $F(m_j)$ returns a value based on whether this element belongs to A_N :

$$F(m_j(ws)) = \begin{cases} 1, & ws \in A_N \\ 0, & ws \notin A_N \end{cases}$$

Given a constant N and a feature m_j , we further derive formula 4.3.2 by counting how many POS tag sequences occurring in the tagged IU that also occur in A_N . Then the re-ranking formula 4.2.1 can be derived into:

$$rel(doc, q) = \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))]$$

$$= \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} + \sum_{ps \in PS(IU_i)} F(m_j(ps)) \right\}$$

After we use this formula to re-ranking the retrieved list of the testing query, we obtain a new MAP of the re-ranked retrieved list. In the same way, we can have a new MAP for each of the other training queries. Finally, we compute the mean of the MAPs of all the queries as the results of the query set with this discourse type.

Table 4.3.4 Cross validation re-ranking retrieval performance based on POS tag 4-grams and feature DF m2

POS Tag Sequence Sorting Feature: DF m2								
Disc Type	baseline	Top N of discourse type related POS tag 4-grams, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.1051	.1604	.1704	.1891	.2153	.2204	.2330
Reason	.2241	.2076	.2246	.2314	.2343	.2416	.2464 [^]	.2448
Impact	.2291	.1866	.2081	.2216	.2308	.2413	.2515	.2556 [#]
Mean	.2219	.1706	.2001	.2105	.2204	.2342	.2413	.2460

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

Table 4.3.5 Cross validation re-ranking retrieval performance based on POS tag 4-grams and feature QF n2

POS Tag Sequence Sorting Feature: QF n2								
Disc Type	baseline	Top N of discourse type related POS tag 4-grams, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.2076	.2200	.2239	.2225	.2254	.2265	.2288
Reason	.2241	.2230	.2417	.2556 [^]	.2493 [^]	.2513 [^]	.2513 [^]	.2510 [^]
Impact	.2291	.2358	.2385	.2398	.2465	.2495	.2515 [#]	.2563 [#]
Mean	.2219	.2240	.2344	.2403	.2407	.2434	.2445	.2471

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

We present the results of the cross-validation experiments based on POS tag 4-grams with feature DF m2 and QF n2 respectively in Table 4.3.4 and 4.3.5. We can see that discourse type “reason” and “impact” are statistically significantly improved with the highest confidence interval based on both features. Most of the queries of discourse type “advantage/disadvantage” can be improved but the highest confidence interval cannot be reached. It means that different discourse types have their own characteristics in the presentation of the discourse types, which can be shown on the different performance resulted from the same discourse type models. Compared with results of using word bigrams with the same features, the results of POS tag trigrams are better, which shows that POS trigram is a better way to detect the discourse type than word bigrams, although word bigrams are better than word sequence with larger length.

In conclusion, the retrospective experiments show that POS 4-grams perform better than POS tag sequences with other lengths. Among all the features, DFR m2 (micro average of the query DFR) and QF n2 (ratio of the sum of query RDFs to the sum of query IDFs) are two best features, which is consistent with word sequences. Cross-validation experiments show that POS tag sequence discourse type model based on POS tag 4-grams has the ability to statistically significantly improve all the queries of discourse types “reason” and “impact” but it cannot consistently improve the queries of “advantage/disadvantage”.

4.4 Discourse Type Based Retrieval by using Word-POS Tag Sequences

Word-POS tag sequences are composed of word(s) and POS tag(s). We will investigate and evaluate Word-POS tag sequences based on Word-POS tag sequence discourse type model. The shortest word-POS tag sequences are a word followed by a POS tag (e.g. *promise IN*) or a POS tag followed by a word (e.g. *NN of*). The length of the shortest word-POS tag sequences is two. A word-POS tag sequence is called POS tag bigram, POS trigram and POS tag N-gram if its length is two, three and N respectively.

A word-POS tag sequences may have the different permutations. For example, word-POS tag bigram has two different permutations: “wp” and “pw”, where “w” denotes a word and “p” denotes a POS tag. Word-POS tag trigram has six permutations: “wpp”, “wpp”, “wpw”, “pwp”, “ppw” and “pww”.

Table 4.4.1 illustrates the same IU that we have taken as examples in the beginning of section 4.2 and 4.3. It was tagged with the POS tagger Monty Tagger [Liu 04] which is a rule-based part-of-speech tagger based on Eric Brill's transformational-based learning POS tagger [Brill 94], and uses Brill-compatible lexicon and rule files. Monty Tagger uses the University of Pennsylvania (Penn) Treebank Tag-set [Santorini 90]. For the reason brevity, we just show some of the POS tag bigrams and trigrams in the table.

Table 4.4.1 An example of an IU and the extracted word-POS tag bigrams and trigram

690 LA031589-0075 the/DT promise/NN of/IN economic/JJ growth/NN and/CC jobs/NNS ,/, but/CC also/RB the/DT chance/NN to/TO develop/VB a/DT unique/JJ society/NN of/IN cultural/JJ ,/, <<educational/JJ >> ,/, technological/JJ and/CC artistic/JJ diversity/NN ./.. To/TO do/VB so/RB ,/, California/NNP must/MD invest/VB now/RB in/IN the/DT necessary/JJ public/JJ support/NN	
Word-POS tag bigrams: "wp" type: the NN promise IN of JJ ... "pw" type: DT promise NN of IN economic ...	Word-POS tag trigrams: "wwp" type: the promise IN, promise of JJ, ... "wpp" type: the NN IN, promise IN JJ, ... "wpw" type: the NN of, promise IN economic, ... "pwp" type: DT promise IN, NN of JJ, ... "ppw" type: DT NN of, NN IN economic, ... "pww" type: DT promise of, NN of economic, ...

Compared with word sequence and POS tag sequence, word-POS tag sequence has larger flexibility and greater ability to define the patterns for text. For example, it's easy to understand that the word-POS tag sequence trigram "more JJ than" can correspond to a comparison structure "more + an adjective + than". In our past work [Wang 06] we discovered that advantages and disadvantages can be derived from comparisons. In that work, the comparative words (e.g. adjective and adverb) and comparison structures (e.g. more... than) are chose as discourse terms. We also found that the higher percentage of comparative words in relevant documents than irrelevant documents for the eight topics that belong to the discourse type advantages/disadvantages. Hence, we find some word-POS tag trigrams that are related with comparison structure or superlative adjective and do some statistics on their distribution in the IUs of the relevant and irrelevant documents. The results are

presented in Table 4.4.2.

Table 4.4.2 Some expressions containing comparative and superlative adjectives and statistics on discourse type “advantage/disadvantage”

Word-POS tag Trigrams	Corresponding Word Sequences	Total RIUF	Total IIUF	IUFR m2	QF n3
VBZ more JJ	is more clever than is more visible than is more plentiful than is more critical of reflects more widespread political is more distressing to	24	4	6.00	4
JJR NN than	stronger bond than any lower overheads than all-purpose deeper recession than most poorer health than those lower rate than ordinary more harm than good	21	8	2.63	4
JJS NN of	fastest growth of real lowest level of health flimsiest understanding of the strongest supporter of SDI strongest criticism of school best example of this	27	8	3.38	6
more JJ than	more clever than us more efficient than private more complex than any more visible than in more successful than any more expensive than in	21	7	3.00	4

Note: the first three words of the word 4-grams matched the word-POS tag sequence and the fourth words are provided only in order to offer more linguistic and background information

Table 4.4.2 shows four word-POS tag trigrams and the some example of their corresponding word sequences are given. These word sequences are selected from the IUs of the retrieved documents of the eight queries of “advantages/disadvantages”. In order to understand these word sequences better, we additionally display the words that are just follow the word trigrams. We counted the numbers of relevant IUs and

irrelevant IUs that containing the word sequences that can be matched by the word-POS tag trigrams, which are shown in the column “Total RIUF” and “Total IIUF”. There are in total 4657 relevant IUs and 3391 irrelevant IUs, the ratio of the two numbers is about 1.37. We can see the ratios of total RIUF to total IIUF (shown in the column “IUF R m2”) are much larger than 1.37. In order to show the consistency, we show the number of queries that satisfy query RDF (relevant document frequency) of this structure is larger than query IDF (irrelevant document frequency), which are shown in the column “QF n3”. It means the expression corresponding to these four types of structure occur more frequently in relevant IUs than the selected irrelevant IUs and this advantage prevails for at least half of the queries (4-6 queries out of 8).

In Table 4.4.3, we give additional two examples that are discovered by statistical work. They both contain an adjective. The number of queries that satisfy query RDF (relevant document frequency) of this structure is larger than query IDF (irrelevant document frequency) is respectively four and five, which are shown in the column “QF n3”. The expressions of the first example use different adjectives to modify the “increase” to show the different ways of “change” (e.g. “*gradual*” or “*substantial*”). The second example surprisingly contains a special noun “system”. From the value of QF n3 feature, we know this structure is consistent. It’s not hard to find that there are two ways to modify “system”. One way is topic entity-related, such as “*a tax-based system*”, “*the anti-missile system*” and “*the educational system*”. The other way is related with time, such as “*the new/old system*”, “*the present system*” and “*the current system*”. We think the difference in time, such as “new/old” and “past/present/future”, can also derive advantages and disadvantage.

Table 4.4.3 Some expressions contain adjective that improve the retrieval of the queries of discourse type “advantage/disadvantage”

Word-POS tag Trigrams	Corresponding Word Sequences	QF n3
a JJ increase	a real increase of a substantial increase in a cost-of-living increase for a modest increase in a near-certain increase in a gradual increase in	4
DT JJ system	a selective system , the dual system , a similar system . the educational system is a tax-based system is the entire system . a tiered system of the national system . a two-tier system of the new system of any other system in the old system on the anti-missile system , the political system in the at-large system of The present system is the contract-based system of the same system . The current system fails the taxation-based system , the current system of	5

Note: the first three words of the word 4-grams matched the word-POS tag sequence and the fourth words are provided only in order to offer more linguistic and background information

Word-POS tag sequence discourse type model is formulated to predict the probability that the discourse type of a part of text (IU, information unit) matches the desirable discourse type based on the word-POS tag sequences of different lengths (e.g. word-POS tag bigrams, trigrams, etc.) that are obtained in their raw text and POS tagged text. A word-POS tag sequence discourse type model can be formulated as:

$$rel(doc, q) = \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))] \quad (4.4.1)$$

In formula 4.4.1, $rel()$ denotes the relevance between a given document doc and query q , $sim()$ denotes the similarity between an IU and a query based on non-discourse type methods, such as vector space model or IU similarity model. $H()$ denotes the probability of IU has the discourse type dt based on the learned function.

“ \wedge ” denotes a conjunctive function. A simple to way to derive $H()$ is as:

$$H(IU_i, dt(q)) = f_{dt}[WPS(IU_i)] = \sum_{wps \in A} tf[wps, WPS(IU_i)] \quad (4.4.2)$$

$wps(IU_i)$ is the data structure consisting of all the word-POS tag sequences with a certain length occurring in IU and their frequencies and positions. $f_{dt}[\]$ is a function obtained from machine learning methods, which determines the probability of the discourse type of an IU, which is presented by a word-POS tag data structure $WPS(IU_i)$, meets the desirable discourse type $dt(q)$ of the query q . One way of learning the function $f_{dt}[\]$ is to learn a set A containing the word-POS tag sequences that are most likely to express the discourse type $dt(q)$. Then all word-POS tag sequences in an IU can be weighted according to whether it belongs to A , or, more complicated, its position in A if all elements are sorted in A . Therefore, we finally derive formula 4.4.2 into the number of word-POS tag sequences that can be extracted from the raw text and tagged text of the IU that occur in A , which is denoted by $tf[\]$.

In next subsection, we will introduce the measures of the distribution of the word-POS tag sequences in the training documents. Some aggregation functions based on these measures provide different ways to derive formula 4.4.2.

4.4.1 The distribution of word-POS tag sequence in relevant documents

Let's assume that the relevant set of a topic presented by query q consists of R documents: $rd_1, rd_2 \dots rd_R$ and the irrelevant set of a topic consists of I documents: $id_1, id_2 \dots id_I$. Let RIU_{ij} be the j -th IU in relevant document rd_i and IIU_{ij} be the j -th IU in

irrelevant document id_i .

Relevant Document Frequency (RDF)

Given a word-POS tag sequence and the raw and tagged text of the IUs of different relevant documents, the number of relevant documents containing the text, in their IUs, that can match the word-POS tag sequence is relevant document frequency, DRF in short. We use $RDF(q, wps)$ to denote the number of relevant documents in the retrieved list of query q containing the text that can match the word-POS tag sequence wps .

Relevant IU Frequency (RIUF)

Relevant IU frequency, denoted by $RIUF$, is the number of the IUs of the relevant documents containing the text that can match a given word-POS tag sequence. We use $RIUF(rd, wps)$ to denote document RIUF, which is the IU frequency of the word-POS tag sequence wps in the IUs of relevant document rd . We use $RIUF(q, wps)$ to denote query RIUF, which indicates the total IU frequency of the word-POS tag sequence wps in the IUs of all the relevant documents of query q . Document RIUF can be normalized by $RIUF(rd, ws)/|rd|_{IU}$, where $|rd|_{IU}$ is the number of IUs that belong to the relevant document rd .

The query RIUF of a word-POS tag sequence is defined based on document RIUF. We can aggregate the document RIUFs of all the relevant documents in the retrieved list in respond to q for a given word-POS tag sequence wps into the query RIUF, which measures overall distribution of the given word-POS tag sequence:

$$RIUF(q, wps) = \underset{i}{agg}[RIUF(rd_i, wps)]$$

A simple way to derive above formula is to use summation to substitute function *agg* and then the result is the total number of IUs that contains *wps* in all the relevant documents in the retrieved list of query *q*:

$$RIUF(q, wps) = \sum_i RIUF(rd_i, wps)$$

Query RIUF can also be computed based on the document RIUF by some more complicated ways, for example, the number of IUs in the relevant document can be considered. Let $R(q)$ be the number of relevant documents for the query *q*. The second way to compute query RIUF is the micro average of percentages of all the IUs that belong to all the relevant documents of the query *q* and contain the text that can match the given word-POS tag sequence *wps*. The micro percentage average weights equally all the documents, regardless of how many IUs belong to it. So we have:

$$RIUF_{micro}(q, wps) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} [RIUF(rd_i, wps) / |rd_i|_{IU}]$$

The third way to compute query RIUF is the macro average of percentages of all the IUs that belong to all the relevant documents of the query *q* and contain the text that can match the given word-POS tag sequence *wps*. We compute the query RIUF by using the macro percentage average as aggregation method as:

$$RIUF_{macro}(q, wps) = \frac{\sum_{i=1}^{R(q)} RIUF(rd_i, wps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}}$$

In summary, given a word-POS tag sequence and a relevant document we propose the concept of document RIUF. Moreover, we propose three ways to compute the query RIUF of a word-POS tag sequence. The simple one is the summation of the RIUF, which is the total number of the IUs in the relevant documents that contain the

word-POS tag sequence. The other two are normalized measures, which are the micro and macro average of the percentages of the IUs of the relevant documents that contain the word-POS tag sequence.

Word-POS tag sequence Frequency (WPSF)

Word-POS tag sequence frequency, WPSF in short, is the number of the occurrence of the word sequences that can match a given word-POS tag sequence in an IU of a relevant document, in a relevant document or in all the relevant documents in the retrieved list of a topic. The WPSF of a word-POS tag sequence in an IU is called IU WPSF. The WPSF of a word-POS tag sequence in a document is called document WPSF. The WPSF of all the relevant documents in the retrieved list of a topic presented by a query is called query WPSF.

We use $freq(RIU_{ij}, wps)$ to denote the IU WPSF of word-POS tag sequence wps in a relevant IU RIU_{ij} (the j -th IU extracted from the i -th relevant document rd_i). Document WPSF is the aggregation of the IU WPSFs of all the IUs that belong to this document. We use $freq(rd_i, wps)$ to denote the document WPSF of the word-POS tag sequence wps in relevant document rd_i , agg_I to denote an aggregating function. So we have

$$freq(rd_i, wps) = agg_I [freq(RIU_{ij}, wps)]$$

A simple derivation of above formula is to substitute function agg_I by summation:

$$freq(rd_i, wps) = \sum_j freq(RIU_{ij}, wps)$$

In above formula, the document WPSF is the total number of occurrence of the word sequences that can match this word-POS tag sequence in all the IUs that belong to the

document.

There is also a problem of normalization for WPSF measures. Obviously, the number of IUs in a document can be used as a normalizing factor. A normalized document WPSF can be obtained by

$$norm_freq(rd_i, wps) = \frac{agg_1[freq(RIU_{ij}, wps)]}{|rd_i|_{IU}},$$

where $|rd_i|_{IU}$ is the number of IUs of the document rd_i . If the aggregating function in above formula is derived by summation, the normalized document WPSF is

$$norm_freq(rd_i, wps) = \frac{\sum_j [freq(RIU_{ij}, wps)]}{|rd_i|_{IU}}$$

The result of above formula is the average IU WPSF of the word-POS tag sequence ps for all the IUs that belong to the document rd_i .

Let a topic be presented by query q . The query WPSF of a word-POS tag sequence is denoted by $Rfreq(q, wps)$. We compute the query WPSF by the following formula where the aggregating function is denoted by agg_2 :

$$Rfreq(q, wps) = agg_2[freq(rd_i, wps)] = agg_2\{agg_1[freq(RIU_{ij}, wps)]\}$$

A simple derivation of above formula is to use summation to substitute both aggregating functions:

$$Rfreq(q, wps) = \sum_i freq(rd_i, wps) = \sum_i \sum_j freq(RIU_{ij}, wps)$$

There are also two ways to compute the query WPSF of a word-POS tag sequence based on the normalized document WPSF of the word-POS tag sequence. The normalized query WPSF of the word-POS tag sequence wps for query q can be obtained by the micro average of the document WPSFs of wps for all the relevant

documents of query q as:

$$\begin{aligned} & norm_Rfreq_{micro}(q, wps) \\ &= \frac{1}{R(q)} \sum_{i=1}^{R(q)} norm_freq(rd_i, wps) = \frac{1}{R(q)} \sum_{i=1}^{R(q)} \frac{\sum_j [freq(RIU_{ij}, wps)]}{|rd_i|_{IU}} \end{aligned}$$

The second way of computing the normalized query WPSF is to calculate the macro average of the document WPSFs of the word-POS tag sequence wps for all the relevant documents of query q :

$$\begin{aligned} & norm_Rfreq_{macro}(q, wps) \\ &= \frac{\sum_{i=1}^{R(q)} freq(rd_i, wps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} = \frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{|rd_i|_{IU}} freq(RIU_{ij}, wps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} \end{aligned}$$

In summary, in the subsection, we put forward the concepts of IU WPSF, document WPSF and query WPSF. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document PSF of a word-POS tag sequence in a document. One is to sum up all the IU WPSFs of the word-POS tag sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU WPSFs of all the IUs that belongs to the document. Second, we propose three ways to compute the query WPSF of a word-POS tag sequence in all the relevant documents of the retrieved list of the query q . They are summation, micro and macro average of the IU WPSFs of the word-POS tag sequence in the IUs that belongs to all the relevant documents.

4.4.2 The distribution of word-POS tag sequence in irrelevant documents

In order to make comparison among the experimental results based on different discourse type models, we use the same way to select the irrelevant documents for word-POS tag sequence discourse type model as we do for word sequence discourse type model and POS tag sequence discourse type model. For each query, we select the same number of the irrelevant documents from the beginning of the retrieved list according to the number of relevant documents of the query. We propose the following measures to reflect the distribution of word-POS tag sequence in the raw and tagged IUs of the selected irrelevant documents.

Irrelevant Document Frequency (IDF)

In the training set of the selected irrelevant documents, the number of the irrelevant documents contain at least one word sequence that can match the word-POS tag sequence is called irrelevant document frequency, IRF. We use $IDF(q, wps)$ to denote IDF of the word-POS tag sequence wps in the retrieved list of query q .

Irrelevant IU Frequency (IIUF)

Irrelevant IU frequency, denoted by IIUF, is the number of the IUs of the irrelevant documents containing at least one sequence that can match a given word-POS tag sequence. We use $IIUF(id, wps)$ to denote document IIUF, which is the IU frequency of the word-POS tag sequence wps in the IUs of irrelevant document id . We use $IIUF(q, wps)$ to denote query IIUF, which denotes the total IU frequency of the

word-POS tag sequence wps in the IUs of all the irrelevant documents of query q . Document IIUF can be normalized by $IIUF(id, wps)/|id|_{IU}$, where $|id|_{IU}$ is the number of IUs that belong to the irrelevant document id .

The query IIUF of a word-POS tag sequence is defined based on document IIUF. We can aggregate the document RIUFs of all the selected irrelevant documents in the retrieved list in response to q for a given word-POS tag sequence wps into the query RIUF. A simple way to derive above formula is to use summation:

$$IIUF(q, wps) = \sum_i IIUF(id_i, wps)$$

Query IIUF can also be computed by using the micro and macro average as aggregation function. Let $I(q)$ be the number of selected irrelevant documents for the query q . We have the following formula by using micro percentage average and macro percentage average:

$$IIUF_{micro}(q, wps) = \frac{1}{I(q)} \sum_{i=1}^{I(q)} [IIUF(id_i, wps) / |id_i|_{IU}]$$

$$IIUF_{macro}(q, wps) = \frac{\sum_{i=1}^{I(q)} IIUF(id_i, wps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}$$

In summary, given a word-POS tag sequence and an irrelevant document we propose the concept of document IIUF. Moreover, we propose three ways to compute the query RIUF of a word-POS tag sequence by using different aggregation functions: summation, micro average and macro average.

Word-POS tag sequence Frequency (WPSF)

There are also three levels of WPSF. The WPSF (word-POS tag sequence frequency)

of a word-POS tag sequence in an IU is called IU WPSF. The WPSF of a word-POS tag sequence in a document is called document WPSF. The WPSF of all the selected irrelevant documents in the retrieved list of a topic presented by a query is called query WPSF.

We use $freq(IU_{ij}, wps)$ to denote the IU WPSF of the word-POS tag sequence wps in irrelevant IU RIU_{ij} . Document WPSF is the aggregation of the IU WPSFs of all the IUs that belong to this document. We use $freq(id_i, wps)$ to denote the document WPSF of the word-POS tag sequence wps in irrelevant document id_i . A simple way to compute document PSF is to aggregate the IU WPSFs by summation:

$$freq(id_i, ps) = \sum_j freq(IU_{ij}, ps)$$

In above formula, the document WPSF is the total number of occurrence of the word sequences that can match this word-POS tag sequence in all the IUs that belong to the document.

There is also a problem of normalization for WPSF measures. A normalized document WPSF can be obtained by

$$norm_freq(id_i, wps) = \frac{agg_j[freq(IU_{ij}, wps)]}{|id_i|_{IU}},$$

where $|id_i|_{IU}$ is the number of IUs of the document id_i . If the aggregating function in above formula is derived by summation, the normalized document WPSF is

$$norm_freq(id_i, wps) = \frac{\sum_j [freq(IU_{ij}, wps)]}{|id_i|_{IU}}$$

The result of above formula is the average IU WPSF of the word-POS tag sequence wps for all the IUs that belong to the document id_i .

Let a topic be presented by query q . The query WPSF of a word-POS tag sequence

is denoted by $Rfreq(q, wps)$. We compute the query WPSF by the following formula based on the aggregating function agg_2 :

$$Rfreq(q, wps) = agg_2[freq(id_i, wps)] = agg_2 \{ agg_1[freq(IIU_{ij}, wps)] \}$$

A simple derivation of above formula is to use summation to substitute both aggregating functions:

$$Rfreq(q, wps) = \sum_i freq(id_i, wps) = \sum_i \sum_j freq(IIU_{ij}, wps)$$

There are also two ways to compute the query WPSF of a word-POS tag sequence based on the normalized document WPSF of the word-POS tag sequence. The normalized query WPSF of the word-POS tag sequence wps for query q can be obtained by the micro average of the document WPSFs of wps for all the selected irrelevant documents of query q as:

$$\begin{aligned} & norm_Rfreq_{micro}(q, wps) \\ &= \frac{1}{I(q)} \sum_{i=1}^{I(q)} norm_freq(id_i, wps) = \frac{1}{I(q)} \sum_{i=1}^{I(q)} \frac{\sum_j [freq(IIU_{ij}, wps)]}{|id_i|_{IU}} \end{aligned}$$

The second way of computing the normalized query WPSF is to calculate the macro average of the document WPSFs of the word-POS tag sequence wps for all the selected irrelevant documents of query q :

$$\begin{aligned} & norm_Rfreq_{macro}(q, wps) \\ &= \frac{\sum_{i=1}^{I(q)} freq(id_i, wps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}} = \frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{|id_i|_{IU}} freq(IIU_{ij}, wps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}} \end{aligned}$$

In summary, in this subsection, we put forward the concepts of IU WPSF, document WPSF and query WPSF to measure the distribution of the word sequences

that can match a given word-POS tag sequence in the IUs of irrelevant documents. We also propose the calculation and normalization methods for these measures. First, we propose two ways to compute the document WPSF of a word-POS tag sequence in a document. One is to sum up all the IU WPSFs of the word-POS tag sequence in all the IUs that belong to this document. The other way is a normalized measure, which considers the number of IUs in the document. The result is the average of the IU WPSFs of all the IUs that belongs to the document. Second, we propose three ways to compute the query WPSF of a word-POS tag sequence in all the selected irrelevant documents of the retrieved list of the query q . They are summation, micro and macro average of the IU WPSFs of the word-POS tag sequence in the IUs that belongs to all the select irrelevant documents.

4.4.3 Distribution features of word-POS tag sequence

In this subsection, we propose some features to evaluate word-POS tag sequence and these features can be used in the later learning procedure. These features are based on the measures on the distribution of the word sequences that can match a given word-POS tag sequence introduced in previous subsections. We also use the ratio of the measure obtained from relevant set to the measure obtained from the corresponding irrelevant set.

Document Frequency Ratio (DFR)

We assume there is a topic set S consisting of T topics that have the same discourse

type and they are presented by queries q_1, q_2, \dots, q_T . We define the query document frequency ratio (query DFR) of a word-POS tag sequence wps for a query as the ratio of the relevant document frequency RDF of the word-POS tag sequence for a query to the irrelevant document frequency IDF to the query, as shown in below formula. A small constant α is to avoid division by zero.

$$DFR(q_i, wps) = \frac{RDF(q_i, wps)}{IDF(q_i, wps) + \alpha}$$

The query set DFR measures the distribution of the word sequences that can match a given word-POS tag sequence in the IUs of the relevant and irrelevant documents of a set of queries. Based on query DFR, we can propose a function to aggregate the query DFRs of all the queries in the query set. It is to aggregate the query DFR of each query in S by an aggregation function $agg(.)$ as:

$$DFR(S, wps) = \underset{q_i \in S}{agg} DFR(q_i, wps)$$

We will further derive above formula in next subsection by substitute the function agg by different operations.

IU Frequency Ratio (IUFR)

IU Frequency ratios are measures based on query RIUF and query IIUF, which can indicate the distribution of the word sequences that can match a word-POS tag sequence wps in the IUs of the relevant and irrelevant documents of a query. We use $IUFR(q_i, wps)$ to denote the query IUFR of the word-POS tag sequence wps for the selected retrieved documents of query q .

The simplest way to compute query IUFR is based on the query IIUF and query

RIUF that are obtained by summing up all the document IIUFs and document RIUFs.

The feature is the ratio of two summations:

$$IUF R(q_i, wps) = \frac{RIUF(q_i, wps)}{IIUF(q_i, wps)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, wps)}{\sum_{i=1}^{I(q_i)} IIUF(id_i, wps)}$$

Query IUF R can use normalized query RIUF and IIUF. For example, the following formula uses query RIUF and query IIUF that come from the macro average of normalized document RIUFs and document IIUFs:

$$IUF R_{macro}(q_i, wps) = \frac{RIUF_{macro}(q_i, wps)}{IIUF_{macro}(q_i, wps)} = \frac{\sum_{i=1}^{R(q_i)} RIUF(rd_i, wps) / \sum_{i=1}^{R(q_i)} |rd_i|_{IU}}{\sum_{i=1}^{I(q_i)} IIUF(id_i, wps) / \sum_{i=1}^{I(q_i)} |id_i|_{IU}}$$

The calculation of query set IUF R is based on the above measures of query IUF R, which indicates the distribution of the word sequences that can match a word-POS tag sequence in the retrieved documents of a set of queries with the same discourse type. In the following formula, $IUF R(S, wps)$ is the query set IUF R of the word-POS tag sequence wps for the query set S , and the $agg(.)$ is an aggregating function.

$$IUF R(S, wps) = \underset{q_i \in S}{agg} IUF R(q_i, wps)$$

We will derive above formula in section 4.3.4 to compute query set IUF R by using different aggregation functions.

Word-POS tag sequence Frequency Ratio (WPSFR)

Query word-POS tag sequence frequency ratio (query PSFR) of a word-POS tag

sequence is based on its query WPSFs in the retrieved documents of the query. For query q and word-POS tag sequence wps , we define query PSFR as the ratio of the query word-POS tag sequence frequency (query PSF) of wps in the relevant documents to the query PSF of wps in the selected irrelevant documents. Let $WPSFR(q_i, wps)$ be the query WPSF of q in the retrieved documents of query q and according to our definition, we have:

$$WPSFR(q_i, wps) = \frac{Rfreq(q_i, wps)}{Ifreq(q_i, wps)}$$

We can derive above formula by using the query WPSF of wps based on summation in the calculation of query WPSFR, we have

$$WPSFR(q_i, ps) = \frac{Rfreq(q_i, wps)}{Ifreq(q_i, wps)} = \frac{\sum_i \sum_j freq(RIU_{ij}, wps)}{\sum_i \sum_j freq(IIU_{ij}, wps)}$$

In above formula, the query WPSFR of q_i is finally derived into the ratio of the summation of the IU word-POS tag sequence frequencies. This is a simple way but not normalized since the number of the IUs in the retrieved documents are is not considered.

We can also derive the query WPSFR by using the normalized query WPSFs based on micro and macro averages measures. The following two formulas show the calculation of the normalized query WPSF.

$$WPSFR_{micro}(q_i, wps) = \frac{norm_Rfreq_{micro}(q_i, wps)}{Inorm_Ifreq_{micro}(q_i, wps)} = \frac{\frac{1}{R(q)} \sum_{i=1}^{R(q)} \frac{\sum_j [freq(RIU_{ij}, wps)]}{|rd_i|_{IU}}}{\frac{1}{I(q)} \sum_{i=1}^{I(q)} \frac{\sum_j [freq(IIU_{ij}, wps)]}{|id_i|_{IU}}}$$

$$WPSFR_{macro}(q_i, wps) = \frac{norm_Rfreq_{macro}(q_i, wps)}{Inorm_Ifreq_{macro}(q_i, wps)} = \frac{\sum_{i=1}^{R(q)} \sum_{j=1}^{rd_i|IU} freq(RIU_{ij}, wps)}{\sum_{i=1}^{R(q)} |rd_i|_{IU}} \cdot \frac{\sum_{i=1}^{I(q)} \sum_{j=1}^{id_i|IU} freq(IIU_{ij}, wps)}{\sum_{i=1}^{I(q)} |id_i|_{IU}}$$

In summary, we propose some features with the ratio form based on the measures of the distribution of the word sequences that can match a word-POS tag sequence which are introduced in last two sections, including DFR which is based on the document frequency of a word-POS tag sequence, IUFR which is based on the IU frequency of a word-POS tag sequence and WPSFR which is based on the word-POS tag sequence frequency. These query-level features indicate the distribution of a word-POS tag sequence in the IUs of the selected relevant and irrelevant documents of a query. In next section, these query-level features will be derived into query set-level features which indicate the distribution of a word-POS tag sequence in all the selected retrieved documents of all the queries with the same discourse type.

4.4.4 Cross validation experiments based on word-POS Tag sequences

In this subsection, we report our experimental results by using a method that is similar with K-fold cross validation to evaluate our word-POS-tag sequence discourse type

model. The method is same with the method used for word discourse type model and POS tag sequence model. So we don't redundantly introduce the method either.

All the word-POS tag sequences with a certain type (e.g “ppw”) connected from the raw and the tagged text of the IUs of the relevant documents are extracted to compose a set A . Every element a_i of set A have a feature set $\{m_j(a_i)\}$ A feature evaluation function $F(.)$ is to evaluation the ability of a_i to improve the retrieval of the training queries based on the feature set of a_i . So we use $F(\{m_j(a_i)\})$ to predict the ability of a_i to improve the retrieval of the testing query. Therefore, in word-POS tag sequence discourse type model we derived formula 4.4.2 into:

$$H(IU_i, dt) = f_{dt}[WPS(IU_i)] = \underset{wps \in WPS(IU_i)}{agg} [F(\{m_j(wps)\})]$$

which shows that we aggregate the function values (by $F(.)$) of each word-POS tag sequence that can match the raw and tagged text of the i -th IU by aggregation function agg .

In order to make a very direct comparison among different features, we propose the function $F(.)$ by the following steps:

(1) According to the given distribution feature m_j , we obtain the values of this feature of all the elements in set A .

(2) We sort all the elements in set A according to the values of their m_j feature.

Then we build a subset A_N of A with the top N elements.

(3) For each element in set A , $F(m_j)$ returns a value based on whether this element belongs to A_N :

$$F(m_j(ws)) = \begin{cases} 1, & ws \in A_N \\ 0, & ws \notin A_N \end{cases}$$

Given a constant N and a feature m_j , we further derive formula 4.4.2 by counting how many word-POS tag sequences that can match the raw and tagged text of the IU that also occur in A_N . Then the re-ranking formula 4.4.1 can be derived into:

$$\begin{aligned} rel(doc, q) &= \underset{i}{agg}[sim(IU_i, q) \wedge H(IU_i, dt(q))] \\ &= \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} + \sum_{wps \in WPS(IU_i)} F(m_j(wps)) \right\} \end{aligned}$$

Table 4.4.4 Cross validation re-ranking retrieval performance based on “pw” type word-POS tag sequences and feature DF m2

Word-POS Tag Sequence Sorting Feature: DF m2								
Disc Type	baseline	Top N of discourse type related “pw” sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.1140	.22	.2215	.2255	.2304	.2433	.2649*
Reason	.2241	.2443	.2561	.2639	.2761	.2794	.2898^	.2957^
Impact	.2291	.1951	.2105	.2178	.2463	.2566	.2598#	.2647#
Mean	.2219	.1880	.2273	.2331	.2498	.2564	.2646	.2744

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

After we use this formula to re-ranking the retrieved list of the testing query, we obtain a new MAP of the re-ranked retrieved list. In the same way, we can have a new MAP for each of the other training queries. Finally, we compute the mean of the MAPs of all the queries as the results of the query set with this discourse type.

Table 4.4.5 Cross validation re-ranking retrieval performance based on “pw” type word-POS tag sequences and feature QF n2

Word-POS Tag Sequence Sorting Feature: QF n2								
Disc Type	baseline	Top N of discourse type related “pw” sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis	.2086	.2575	.2313	.2318*	.2333*	.234*	.2365*	.2369*
Reason	.2241	.2494	.2557	.2548^	.2548^	.2557^	.2582^	.2624^
Impact	.2291	.2559	.2571	.2569#	.2584#	.2586#	.2597#	.2608#
Mean	.2219	.2543	.2495	.2493	.2504	.2509	.2528	.2547

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

We present the results of the cross-validation experiments based on word-POS tag sequence type “pw” with feature DF m2 and QF n2 respectively in Table 4.4.4 and 4.4.5. We can see that all the three discourse types are statistically significantly improved with the highest confidence interval based on both features. Results of feature DF m2 have bigger mean MAPs for each run than QF n2. The mean MAP of all the queries of the three discourse types by using DF m2 feature can be improved by 0.05. However, QF n2 feature can make the significantly improved results appears with fewer “pw” type sequence. Compared with results of using word bigrams and POS tag trigrams with the same features, the results of word-POS tag sequences with “pw” are better because “pw” sequences produce significantly improved results for all the discourse types.

In conclusion, the retrospective experiments show that “pw” type word-POS tag bigrams perform better than word-POS tag trigrams. Among all the features, DFR m2 (micro average of the query DFR) and QF n2 (ratio of the sum of query RDFs to the sum of query IDFs) are two best features, which is consistent with word sequences and

POS tag sequences. Cross-validation experiments show that word-POS sequence discourse type model with “pw” type sequences has the ability to statistically significantly improve all the queries of all the three discourse types.

4.5 Comparison and Analysis of the Different Types of Linguistic Sequence and Features Evaluation

4.5.1 Analysis of the different types of linguistic sequences

From last three sections, we report the results of the retrospective and cross-validation experiments based on word sequence discourse type model, POS tag sequence discourse type model and word-POS tag sequence discourse type model with different sequence selection features. We have made use of different types of sequences to improve the retrieval for the selected discourse types. In this section we will deeply analyze the different types of sequences, analyze the same type of sequences with different lengths and then we make comparison among all the sequences. Also, we will try to make use of combinations of different types of sequences like a cocktail treatment to see whether we can achieve even better performance.

Based on the above experiments, it is known that we can use both retrospective and cross validation experiments to evaluate different linguistic sequence features because they can both identify the best feature. For example, feature DF m2 is better than DF m1 and m3 in the retrospective experiments and this is also true for the cross validation experiments. All the three discourse types support this conclusion. Therefore,

the retrospective results presented in this section will make us draw the same conclusion as cross validation experiments.

We also found that the evaluation of the types of linguistic sequences and features does not rely on the discourse type. The evaluation results are general for all the three discourse types. The retrospective experimental results in appendices A1, A2 and A3 and cross validation results in section 4.2.4, 4.3.4 and 4.4.4 show that, given a specific type of linguistic sequence, all the three discourse types consistently justify the same evaluation result of the features. For example, in Table 4.4.4, “pw” type sequences are used and DF m2 is the best feature for all the three discourse types. Moreover, for a given type of linguistic sequences, all the three discourse types consistently have the same evaluation result for sequences with different lengths. For example, word bigram produces the best results among word sequences with different length (bigram, trigram, 4-gram) and this conclusion is applicable for the three discourse types. Therefore, we can say that the evaluation results of different types of sequences and features are independent of discourse type and all the three discourse types support the same conclusion.

By using the same evaluation feature, we are able make comparison among different types of linguistic sequence and attempt to find the potential reasons for their performance difference. For brevity, we will randomly select a discourse type and we can draw universal conclusions based on the retrospective result.

Generally speaking, word bigram produces the best results among word sequences with different length; POS tag 4-gram produces the best results among POS tag sequences with different lengths; “pw” type word-POS tag sequence produces the best

results among the word-POS tag sequences with different lengths and types.

In order give a clear view on the results of different discourse types models, we list the retrospective re-ranking results for nine queries of discourse type “impact” based on feature DF m2 and QF n2 with 500 and 1000 word/POS tag/word-POS tag sequences in Table 4.5.1.

Table 4.5.1 Comparison of the results from different discourse type retrieval models

Discourse Type: Impact (baseline=0.2241)					
Feature		DF m2		QF n2	
N of A_N		500	1000	500	1000
POS Tag Sequence	bigram	0.2389	0.2241	0.2260	0.2241
	trigram	0.4012*	0.4200*	0.4010*	0.4173*
	4-gram	0.4428*	0.4889*	0.4368*	0.4654*
	5-gram	0.4673*	0.4902*	0.4428	0.4758
Word Sequence	bigram	0.3187	0.2964 [^]	0.3377	0.3650
	trigram	0.3130	0.3896	0.3178	0.3842
	4-gram	0.2753	0.3423	0.3484	0.4318
Word-POS Tag Sequence	pw	0.4980*	0.5407*	0.4240	0.4989*
	ppw	0.4939*	0.5426*	0.4641*	0.5082*
	wwp	0.4284	0.5054	0.436	0.5281
	pwp	0.4930*	0.5210*	0.4316	0.4888*
	wpw	0.4916*	0.526*	0.4431	0.5106*

[^] and * respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22% and 99.61%.

Let us look at POS Tag sequence. Obviously, POS tag 4-gram is the best on among all the POS tag sequences in terms of mean MAP and significance tests. POS tag 5-gram is better the 4-gram in mean MAP but cannot produce significantly improved results for QF n2 feature. POS tag 4-gram is better than trigram due to higher mean MAPs. This conclusion is consistent with the overall experimental results of all the three discourse types.

For word sequence, word bigram is best because it have the only run at feature DF

m2 and N=1000 being significantly improved at 99.61% confidence. The results of word trigram are better than word 4-gram in terms of mean MAP. This conclusion is consistent with the overall experimental results of all the three discourse types.

For word-POS tag sequence, “ppw” is the best one in terms of significant tests. Obviously, the second best one is “pw” and “wwp” is the worst one. We find that the differences among the word-POS tag sequences with different types are much less than the difference among word sequences and POS-tag sequences. This conclusion is not consistent with the overall experimental results of all the three discourse types. We find that “pw” is better than “ppw” in Section 4.4. We think it’s because the performances of “pw” and “ppw” are quite close.

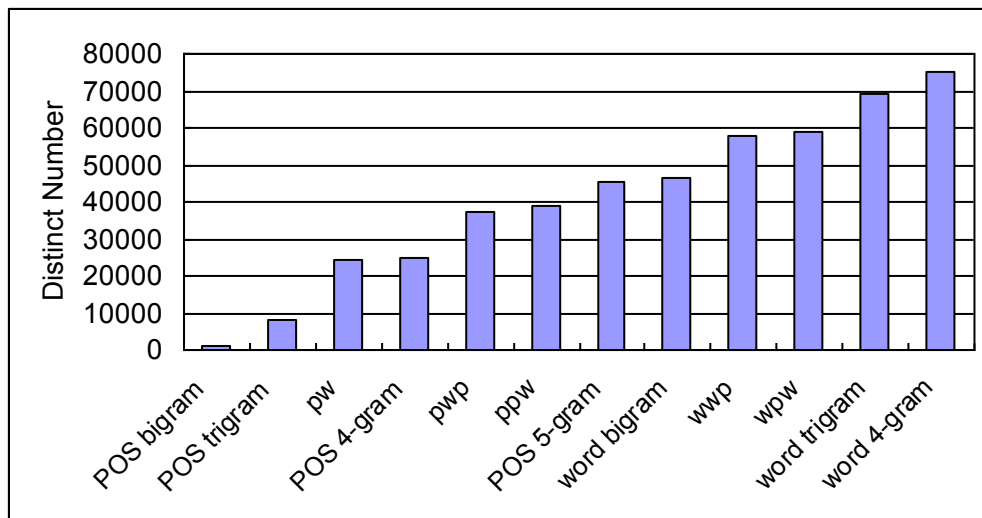
If we make a comparison among all the sequences, word-POS tag sequences are generally better than POS-tag sequences, which are better than word sequences. POS 4-gram and 5-gram are comparable with “pw” and “ppw” type of word-POS tag sequences.

In fact, these sequences are different presentations of the same text by consistently substituting some group of words by the same POS tag. So we will study each type of these sequences for the characteristics of their occurrences.

First let us look at the distinct number of different types of sequences. These sequences are extracted or jointly extracted from the raw and tagged text of the IUs. Also, we explained that from a 41-word IU, there are totally 40 bigrams that can be possibly extracted and 39 trigram and 38 4-grams. So the total numbers of different types of the sequences are more or less the same. But obviously the distinct numbers are quite different. Since the total number are. Less the distinct number, more likely

they occur repeatedly or less specialty. We do a statistical experiments on all the sequences extracted from 3267 relevant IUs of discourse type “impact”. There are 130,680 bigrams and 120,879 5-grams that can be possibly extracted. In Figure 4.5.1, we illustrate the distinct numbers of different types of sequences. We sort the types in ascending order according to the distinct numbers from left to right.

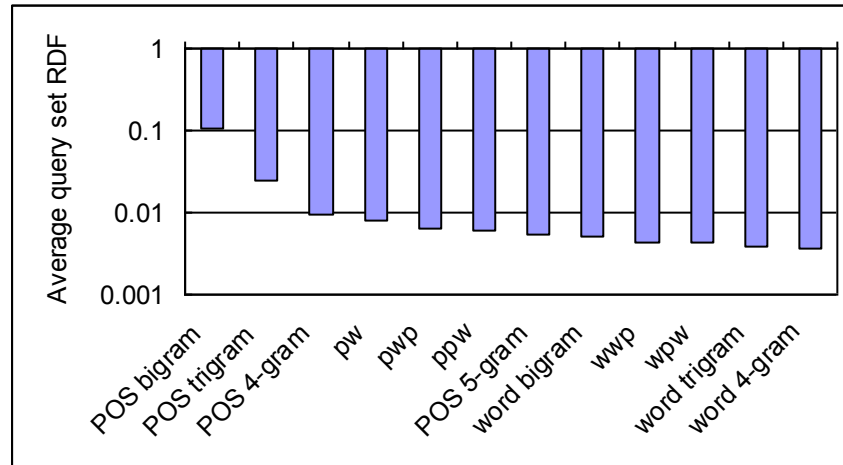
Figure 4.5.1 Distinct number of different types of sequences.



From Figure 4.5.1, we can see that in terms of distinct number, “pw” types sequences and POS tag 4-gram are very close. So are “pwp” and “ppw” types, POS tag 5-gram and word bigrams, “wwp” and “wpw”. It’s difficult to deduce a range based on distinct number that can produce good retrospective results, since between 20000 and 60000, there located very good sequences from Table 4.5.1.

A query set RDF is the number of documents containing the text, in their IUs, that can match the given sequence of all the entire retrieved relevant documents. So query set RDF is a good measure of specialty for a sequence. We compute the average query set RDF for all sequences of a type and put them in Figure 4.5.2 in descending order.

Figure 4.5.2 Average query set RDF of different types of sequences.



The ranks of most types in Figure 4.5.2 conforms to the ranks in Figure 4.5.1 because when distinct number is less and the total number keeps unchanged the occurrence of each sequence is more frequent and the RDF of this sequence will be higher. However, the differences that lie in the average query set RDFs of these types are still not big enough. It's because more of the sequence occur only in one IU, which is in accordance with Zipf's law. So we will study a set of sequences as a whole, and we also try not to be disturbed by the large percentage of low-frequency sequences.

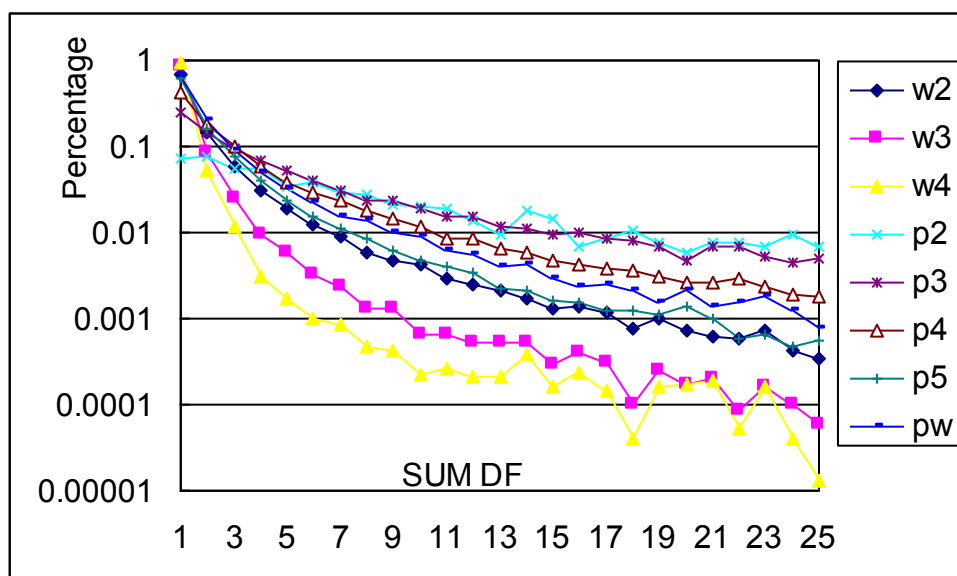
In order to study the differences in performance, we study the distribution of the sequences of each type according to their Zipf's curve. Different types have different curves in the shape but it's difficult to compared the shape of Zipf's curves since the number of sequences of all the types are quite different. We propose an alternative way. We investigate of how many sequence of a given type have a certain document frequency (DF) instead of occurrence frequency. We have 460 relevant and irrelevant documents for nine queries of "impact" and for each type. So the maximum DF is 460 and minimum is 1. For each possible value, we count the number of the sequences. We use percentages rather than absolute numbers as a normalization way.

Table 4.5.2 Percentages of each types of sequences that have a certain DF value

SUM DF	Word Sequence			POS Tag Sequence		
	bigram	trigram	4-gram	bigram	trigram	4-gram
1	68.88%	85.87%	92.36%	7.20%	25.07%	43.25%
2	14.19%	8.61%	5.33%	7.68%	14.18%	17.87%
3	5.82%	2.53%	1.15%	5.47%	9.36%	9.90%
4	3.06%	0.93%	0.31%	5.47%	6.73%	5.85%
5	1.85%	0.58%	0.17%	3.36%	5.17%	3.89%
10	0.43%	0.07%	0.02%	2.02%	1.92%	1.19%
20	0.07%	0.02%	0.02%	0.58%	0.47%	0.26%
50	0.01%	0.0029%	0.0027%	0.29%	0.14%	0.05%

In Table 4.5.2, we present some results of the statistics on the percentages of sequences and some DF values. For example, 68.88% of the word bigrams occur in the IUs of only one document and 0.43% of them occur in ten documents. 7.2% of POS tag bigrams occur in one document and 2.02 of them occur in ten documents.

Figure 4.5.3 Percentages of each types of sequences that have a certain DF value (part I)



Note: w2-4 denote word bigram, trigram and 4-gram; p2-5 denote POS tag bigram, trigram, 4-gram, 5-gram.

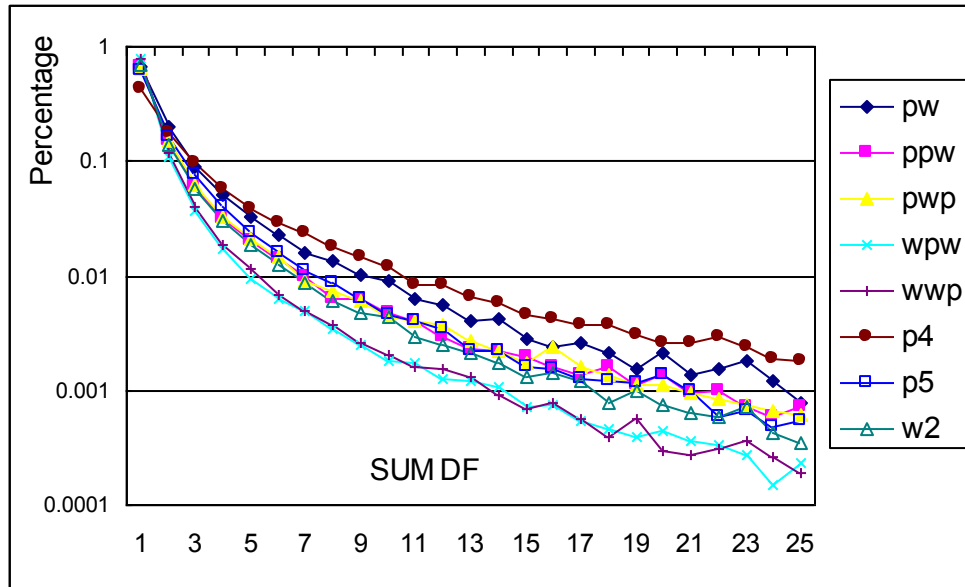
After we do these statistical works on all the types of sequences, we present the percentages of the DF values ranging from 1 to 25. First, these statistical data have bigger difference when DF values are small. Second, we find that for the numbers

larger than 25, some types, especially the word sequences, do not contain any sequence that have the same DFR value. So we cannot compare the percentages at these numbers.

Figure 4.5.3 (part I) illustrates the broken lines of the percentages of some types of sequences at each DF value between 1 to 25. We can see the two lines on the top are POS tag bigram and trigram and two lines at the bottom are word trigram and word 4-gram. The two groups have bad performance in the retrospective experiments. In the middle, lines of POS 4-gram (p4, triangle) and POS 5-gram (p5, cross) are two boundaries and the types with the good performance lie between the boundaries.

Let us have a more detailed view on the area between POS 4-gram and POS 5-gram. We put the other types of word-POS tag sequences in a separate figure. From part II of Figure 4.5.3, we find that “wpw” and “wwp” types have very close lines at the bottom. POS tag 4-grams lie on the top and “pw” sequence lies just under it. “ppw”, “pwp” types and POS tag 5-gram are very close to distinguish and word bigram lie at the bottom of above three’s overlapped lines. From the experimental results shown in the Table 4.5.1, we conclude that the lines that lie around POS tag 4-gram and “pw” have steadily good performance. They point the approximate area that can produce good performance in our figure. Area around “ppw” is hybrid, which includes the sequence that performs quite well such as “ppw”, the sequence that performs ordinary such as POS tag 5-gram and the sequence that performs not so well, such as word bigram.

Figure 4.5.3 Percentages of each type of sequences that have a certain DF value (part II)



In conclusion, our normalized Zipf’s curve of sequence’s DF can indicate the overall distribution of the sequences with the same type based on their likelihood to occur in more documents. This first principle for mining discourse type related sequences is that they occur in more documents. The second principle is that they occur differently in relevant and irrelevant documents. The query level measures and query set level features proposed and used in our discourse type models intend to find the difference of the occurrence of a sequence in relevant and irrelevant documents, which is to quantify the requirement of the second principle. The normalized Zipf’s curve is to investigate which sequence type has a large percentage sequences that are likely to occur in more documents, which is to quantify the requirement of the first principle. Our normalized Zipf’s curve can reflect the quality of sequence for detecting and presenting discourse types. The “pw” type word-POS tag sequence is the best sequence we’ve found based on our study. Closer to the normalized Zipf’s curve of “pw” type sequence, the better respective and actual re-ranking results a sequence type

can produce.

4.5.2 Combinations of the different types of linguistic sequences and Feature Evaluation

After evaluating and comparing all types of the linguistic sequences, we are interested to know whether the combinations of different types can also enhance the retrieval. If two or more types of linguistic sequences are used, can it enhance the retrieval more than the single type? Hence, we choose some top-ranked types in the later experiments.

We setup experiments in order to see whether adding a different type of sequences can improve the results of using only the best types. We have shown that “pw” type is the best type in terms of the retrieval performance. The retrospective experiments are performed to give the general view of the performance of the different sequences sets and our abundant experiments have shown that the performance rank of cross validation experiments is always the same as the retrospective experiments.

Group 1 tests 100 linguistic sequences. No.101 uses top 100 “pw” type (POS tag + word) sequences selected from the sorted list (sorted by one of the best sorting measures: QF m2). We combine the other types of linguistic sequences with “pw” type in the later runs. No.102 uses top 50 sequences from the above mentioned “pw” type list and another top 50 from the “pp”(POS tag bigram) list which has been sorted by the same measure. No. 103 uses top 50 sequences from the “pw” type list and another top 50 from the “wwp”(word+word+POS tag) list sorted by the same measure. From the former experiments we know that “wwp” type is the second best type after “pw”

type”. No 104 uses “pwp” type and No.105 use the mixture of the three best types.

Table 4.5.3 Retrospective performance in MAP of using combination of different types of word-POS Tag Sequences (Discourse type: reason)

Part I

Group 1: 100 sequences		
No.101	top 100 "pw"	0.2818
No.102	top 50 pw + top 50 pp	0.2681
No.103	top 50 pw +top 50 wwp	0.2796
No.104	top 50 pw+ top 50 pwp	0.2701
No.105	top 33 pw+ 33 pwp + 33 wwp	0.2714
Group 2: 200 sequences		
No.201	top 200 "pw"	0.3498
No.202	top 100 pw +top 100 pp	0.3233
No.203	top 100 pw +top 100 wwp	0.3362
No.204	top 100 pw+ top 100 pwp	0.3221
No.205	Top 66 pw+ 66 wwp+66 pwp	0.3337

Part II

Group 3: 1000 sequences		
No. 301	top 1000 "pw"	0.4989
No. 302	top 500 pw +top 500 pp	0.2374
No. 303	top 500 pw +top 500 wwp	0.4762
No. 304	top 500 pw+ top 500 pwp	0.4742
No. 305	top 333 pw+ 333 pwp + 333 wwp	0.4663
Group 4: setting threshold		
No.401	Threshold: QF n2 >=40 100 sequences (pw, pwp, wwp) (100 is just coincident)	0.2799
No.402	Threshold: QF n2 >=30 615 sequences (pw, pwp, wwp)	0.4374

The conclusion drawn from Group 1 is that if the same number of sequences are used, only use “pw” types is better than combining ”pw” type with other types, even the second best type “wwp”. The Group 2, 3 and 4 also support the above conclusion with the total number of the sequences increases. By comparing No.101 with No.202, 203, we find that adding more top ranked sequences (e.g. 100 “pp” and 100 “wwp”

sequences) in different types improve the results. However, adding 100 top-ranked “pp” and 100 top-ranked “wwp” sequences are both worse than adding 100 “pw” sequences ranked from 101 to 200, which is just No.201 run.

More specific, by comparing No.201 and 203, we find that the second best part (which rank from 101 to 200) of the best type (“pw”) is even more helpful than the best part (top 100) of the second best type (“wwp”). In addition, No.301 outperforms 303, which also supports this conclusion. This shows that the big differences exist among the abilities of different types of linguistic sequences to enhance the retrieval. Group 4 selects the linguistic sequences by setting a threshold on a feature (QF n2) and it does not fix the number of each type. The conclusion drawn from Group 4 is consistent with the previous groups. The above experiments are based on the discourse type “reason” and the experiments on the other two discourse types “advantages/disadvantage” and “impact” result in the same conclusion.

The cross-validation experiments also support the above conclusion and we present the results in Table 4.5.4.

The reason of the above phenomenon is that the best type “pw” has the best ability to represent the good word sequences that can empirically improve the retrieval because it has the optimal specificity. A more special representation (e.g. “wwp” type) may miss some useful sequences and a more general representation (e.g. POS tag bigram) may bring more dirty sequences.

It is obviously that a word and a POS tag generally have quite different specificities which results in the big specificity difference in different types of sequences, such as “wwp” and “pw”. Hence they should not be used in parallel to

detect potential text expressions that improve retrieval, let alone to use them one followed the other. When “wwp” is very specific for detecting, “wwp” followed by a “pw” will be more specific than “wwp”, so that it cannot be helpful.

Table 4.5.4 Cross validation performance in MAP of using combination of different types of word-POS Tag Sequences (Discourse type: reason)

Group 1: 100 sequences		
No.101	top 100 "pw"	0.2557
No.102	top 50 pw +top 50 wwp	0.2330
No.103	top 33 pw+ 33 pwp + 33 wwp	0.2261
Group 2: 200 sequences		
No.201	top 200 "pw"	0.2548
No.202	top 100 pw +top 100 wwp	0.2401
No.203	Top 66 pw+ 66 wwp+66 pwp	0.2381
Group 3: setting threshold		
No.301	Threshold: QF n2 >=40 100 sequences (pw, pwp, wwp) (100 is just coincident)	0.2264
No.302	Threshold: QF n2 >=30 615 sequences (pw, pwp, wwp)	0.2447

We propose several features (e.g.QF n2) to evaluate the performance of the different types of linguistic sequences and in this section we make a comparison between current state-of-the-art measure (w4 in [Robertson 76] as example) and our features.

In the retrospective experiments, as for the same feature (e.g. QF n2), the unbalanced one is better than the balanced one because the balanced one depends on more training documents. For the unbalanced retrospective results, w4 is better than DF m2 because DF m2 is only based on the number of relevant document and irrelevant documents that containing a linguistic sequence (they are r and $n-r$ in w4 formula) and w4 depends on r , $n-r$, $R-r$ (number of relevant documents that do not

contain the linguistic sequence), $N-n-R+r$ (number of irrelevant documents that do not contain the linguistic sequence). For the unbalanced retrospective results, QF m2 is better than w4, which shows that QF m2 is a better feature than w4 in term of selecting the infrequent linguistic sequences that can help to distinguish relevant from irrelevant documents.

Table 4.5.5 Retrospective performance in MAP of using combination of different types of word-POS Tag Sequences (Discourse type: reason)

	Balanced QF n2	unbalanced QF n2	unbalanced DFm2	unbalanced w4
Group 1: 100 sequences				
top 100 "pw"	0.2818	0.3692	0.3071	0.3236
top 50 pw +top 50 wwp	0.2796	0.4606	0.3383	0.3483
top 50 pw+ top 50 pwp	0.2701	0.4298	0.3106	0.3222
Group 2: 200 sequences				
top 200 "pw"	0.3498	0.4896	0.4262	0.4384
top 100 pw +top 100 wwp	0.3362	0.5849	0.3872	0.3994
top 100 pw+ top 100 pwp	0.3221	0.4988	0.3889	0.4006

Table 4.5.6 Cross validation performance in MAP of using the different measures (Discourse type: reason)

Baseline=0.2241	Balanced QF n2	unbalanced QF n2	balanced w4	unbalanced w4
Group 1: 100 sequences				
top 100 "pw"	0.2557	0.1933	0.2234	0.1983
top 50 pw +top 50 wwp	0.2330	0.1834	0.2135	0.1842
Group 2: 200 sequences				
top 200 "pw"	0.2548	0.1954	0.2244	0.2035
top 100 pw +top 100 wwp	0.2401	0.1815	0.2154	0.1823

In the cross-validation results, by using the same feature, balanced training is better than unbalanced training. Apparently, the small numbers of high-ranked irrelevant documents are better negative instances than the large number of low-ranked irrelevant documents. High-ranked irrelevant documents always contain adequate topic

entity terms but they lack discourse related sequences, as a result, they are not relevant. So theoretically, they are good negative instances. On the other hand, empirically, to lower the ranks of high-ranked irrelevant documents (is equal to lifting the rank of a high-ranked relevant document) improves the MAP more significantly than lowering low-ranked irrelevant documents.

In conclusion, the different types of linguistic sequences and the different formats of the same linguistic sequences (such as “pw” and “ppw” types of word-POS tag sequences) have quite different abilities to enhance the discourse type based retrieval, which causes that the combinations of different types (formats) of linguistic sequences cannot outperform the best type (format). Second, our features (e.g. QF n2, the ratio of the sum of query RDFs to the sum of query IDF_s) are better than the state-of-the-art feature w4 in the IU-based retrieval. It is because our features have more ability to select the good linguistic sequences that occur infrequently which is always be ignored by a large-scale statistics.

Summary

In this Chapter, we propose the concept of discourse type retrieval and we choose typical and appropriate examples from TREC topic to study. By manually checking the TREC robust track topics, we discover three discourse types for research: “advantage/disadvantage”, “reason” and “impact”. Each discourse type has its own character to be presented by languages. We choose these discourse types because the

number of topic is abundant for machine learning and the performance of traditional retrieval is below average.

We use different types of linguistic sequences to support the discourse type based retrieval and these sequences include words (e.g. “*strongest criticism*”), POS tags (e.g. “*JJS NN*”, a superlative adjective + a noun) and mixture of both (e.g. “*JJS NN of*”). We detect the discourse type information by measuring the distribution of linguistic sequences in relevant and irrelevant instances (IUs). It is difficult to select the negative instance: the text lack of discourse type information. An irrelevant IU is a good negative instance because it contains topic entity information so it is irrelevant due to the lack of discourse type information. That is why we put the discourse type retrieval under the background of IU-based retrieval.

By retrospective and cross-validation experiments, we can make comparison among performances of the different types of linguistic sequences. Different types of linguistic sequences can be applied independently or altogether to enhance retrieval. Word sequences are too specific to be matched and POS tag sequences are so general to lose the ability of accurate matching. Hence, the Word-POS tag sequences are compromise of the two and have more powerful ability to enhance the retrieval by using the appropriate measures. The word-POS tag sequence may have different formats and “pw” format (a POS tag followed by a word, or “wp” format) gave the best performance for discourse type based retrieval in our experiments. Compared with popular measures such as w4, our proposed features are very helpful in choosing helpful linguistic sequences, although they are very simple.

CHAPTER 5

APPLICATION OF PATTERN RECOGNITION TECHNOLOGIES IN DISCOURSE TYPE BASED INFORMATION RETRIEVAL

5.1 Introduction

The appearance and application of pattern recognition technologies can be traced back to the middle of the 20th century. The rapid development of the computer hardware and software technologies moved the pattern recognition from a theoretical research in the field of statistics to practical applications. Nowadays, automation in industrial production and the need for text and multimedia information processing are becoming more and more important and this trend makes pattern recognition one of the most useful methods that are deeply investigated and widely exploited in the engineering applications and research. Pattern recognition is a very challenging and multidisciplinary research area that attracts researchers and attention from a lot of fields, including computer science, artificial intelligence, statistics, medical science and forensic analysis, etc.

The goal of pattern recognition can be simplified into the classification of objects

into a number of classes. These objects can be images, text or any type of measurements depending on different application, which are referred to as “patterns”. Pattern recognition aims to classify the patterns based either on a priori knowledge or on statistical information extracted from the patterns. A complete pattern recognition procedure consists of the collection of observations to be classified or described, a feature extraction mechanism that computes numeric or symbolic information from the observations, a feature selection mechanism to select the best features to generate the best performance of the learning model and a classification or description scheme that does the actual job of classifying.

The classification scheme is usually based on the availability of a set of patterns that have already been classified. This set of patterns is called the training set, and the later learning strategy based on the training set is characterized as supervised learning. Learning can also be unsupervised, if the system is not given the labels of patterns, instead the learning scheme itself establishes the classes based on the statistics of the patterns. In our experiments, we use supervised learning methods.

The classification scheme usually uses one of the following approaches: statistical (or decision theoretic) or syntactic (or structural). Statistical pattern recognition is based on statistical characters of patterns, assuming that the patterns are generated by a probabilistic model. Syntactical (or structural) pattern recognition is based on the structural interrelationships of features. A wide range of algorithms can be applied for pattern recognition, from very simple Bayesian classifiers to much more powerful neural networks.

5.2 Feature Construction and Selection

5.2.1 Vector formulation

Let S is a set of distinct linguistic sequences $\{s_1, s_2, \dots, s_N\}$ and S can be a set of word sequences, POS tag-sequences or word-POS tag sequences. The cardinality of set S is N . We formulate an IU as a vector $X = \{x_1, x_2, \dots, x_N\}$ and each element x_i in X corresponds to the frequency of the linguistic sequence s_i that can be matched in raw and tagged text in the IU. The label of vector X is denoted as $L(X)$ and $L(X)$ has two possible nominal values R and I (or 1 and -1) which respectively indicate the IU presented by X is relevant or irrelevant to the corresponding topic. A training set is a set of vectors $\{X_1, X_2, \dots, X_m\}$ and their known labels $\{L(X_1), L(X_2), \dots, L(X_m)\}$. A testing set is a set of vectors $\{X_{m+1}, X_{m+2}, \dots, X_{m+n}\}$. A learning model is to predict the relevance status of these vectors based on the information learned from training set.

5.2.2 Feature selection

Feature selection, also known as variable selection or feature reduction, is the technique, commonly used in machine learning, of selecting a subset of relevant features for building robust learning models. Given a number of features, we need to reduce their number and at the time try to keep as much as possible of their discriminatory information. We know that if we selected features with little discrimination power, we cannot expect the later classifier to have good performance.

On the contrary, if good features are selected, the design of classifier can be greatly simplified. By removing most irrelevant and redundant features from the data, feature selection helps to improve the performance of learning models by alleviating the effect of the curse of dimensionality, enhancing generalization capability, speeding up learning process and improving model interpretability. Feature selection also helps people to acquire better understanding about their data by telling them which are the important features and how they are related with each other.

The number of features related with our application is usually very large and this number can easily become of the order of hundreds. Computational complexity is one of the obvious reasons for the necessity to reduce the number of features. It's possible to reduce the number of features because some of them are mutually correlated. We also know that the higher the ratio of the number of training instances to the number of free classifier parameters, the better the generalization power of the classifier. The number of classifier parameters depends on the number of features. So when we have limited training data, we need to reduce the number of features to provide a classifier with more powerful generalization ability.

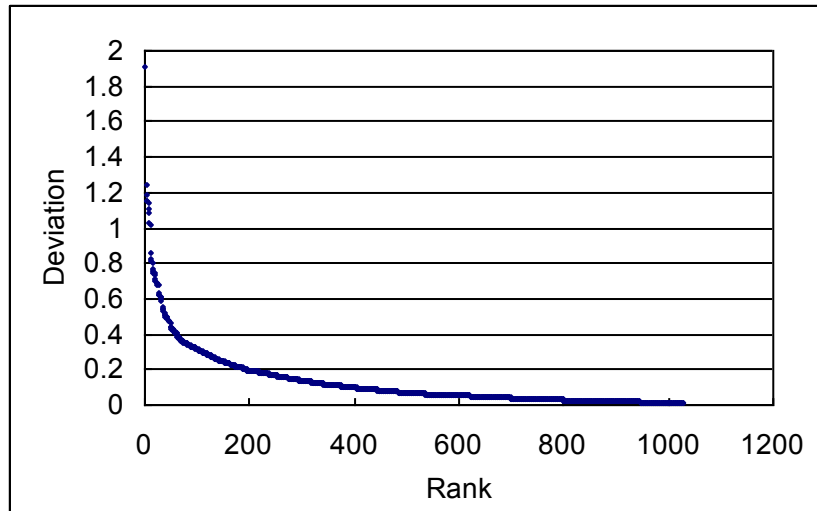
To remove useless attributes is one of the very simple feature preprocessing methods. It simply removes such attributes only providing a very small deviation. Given a threshold, we can remove those features which have the same or smaller deviation than the threshold among all the values in the patterns. Let the values of the i -th feature \mathbf{x}_i of the m patterns are respectively $x_{i1}, x_{i2}, \dots, x_{im}$. We estimate the standard deviation δ_i of the possible values of feature \mathbf{x}_i by regarding the existent data as examples:

$$\delta_i = \sqrt{\frac{\sum_{j=1}^m (x_{ij} - \bar{x})^2}{(m-1)}}$$

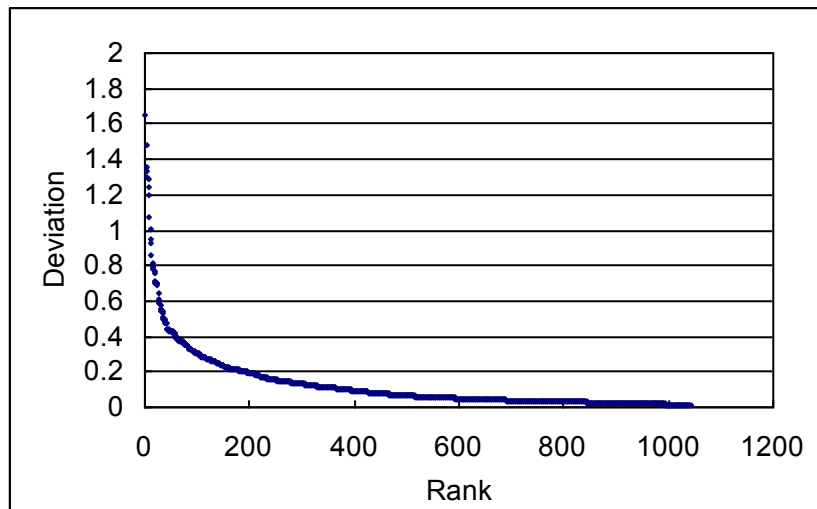
where \bar{x} is the sample average and m is sample size.

In figure 5.2.1 we illustrate the standard deviations of the feature values related with all the POS tag bigrams of the three discourse types. Each POS tag bigram corresponds to a feature and the number of its occurrence in an IU is the feature value. Each selected relevant IU or irrelevant IU is regarded as a pattern. We estimate the standard deviation of the feature based on the patterns we have according to above formula. Then we rank the standard deviations in descending order. The numbers of features for the three discourse types are very close. From the diagrams we find that the distributions of the points in three diagrams are also very alike. The features with very small deviations correspond to the very frequently used or very rarely used POS tag bigrams. From the Zipf's law, we know most of above-mentioned features are rarely used features. We can filter these features out by setting a threshold for deviation.

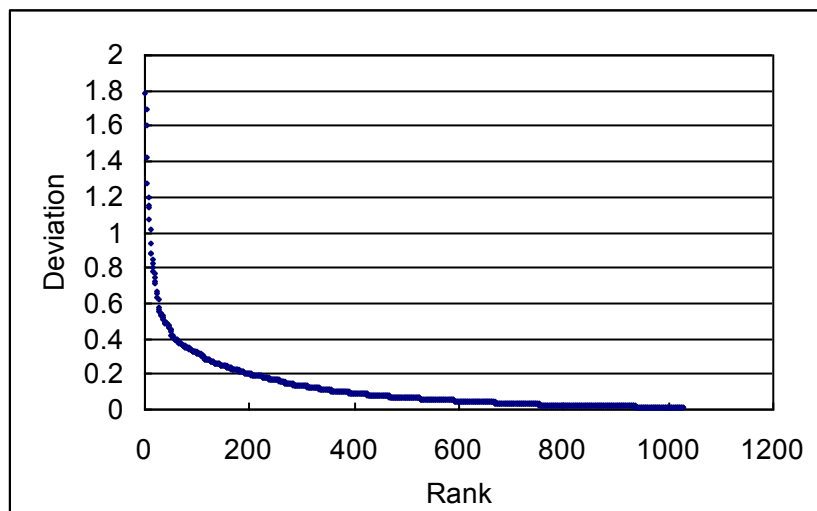
Figure 5.2.1 Ranked estimate standard deviations of all the features of POS tag bigrams



(Discourse type: advantage/disadvantage. Number of features: 1044. Number of patterns: 11193)



(Discourse type: reason. Number of features: 1042. Number of patterns: 6094)



(Discourse type: impact. Number of features: 1030. Number of patterns: 8612)

5.2.3 Feature space transformation

Principal component analysis (PCA) is a vector space transform often used to reduce multidimensional data sets to lower dimensions for analysis. Depending on the field of application, it is also named the discrete Karhunen-Loève transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD). PCA is the simplest of the true eigenvector-based multivariate analyses.

PCA [Jolliffe 02] is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for a given data in least square terms.

PCA can be used for dimensionality reduction in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data.

Suppose we have N IUs for training and each of IU has M features, and we want to reduce the features so that each IUs can be described with only L features, $L < M$. Suppose further, that the IUs are arranged as a set of N data vectors x_1, x_2, \dots, x_N with each x_n representing a single grouped observation of the M features for the n -th IU.

We write x_1, x_2, \dots, x_N as column vectors and each vector has M rows. Then we put the column vectors into a single matrix X of dimensions $M \times N$. According to the symbol shown in Table 5.2.1, we perform PCA on the data set with L feature in the following

steps:

Table 5.2.1 Symbols used in this section and their meanings

Symbol	Meaning
$X = \{X[m, n]\}$	original data matrix
$u = \{u[m]\}$	vector of empirical means for the rows in X
$s = \{s[m]\}$	vector of empirical standard deviations for the rows in X
$h = \{h[m]\}$	vector consisting of 1's
$B = \{B[m, n]\}$	deviation matrix consisting of the deviations from the mean of each row of X
$C = \{C[p, q]\}$	covariance matrix
$R = \{R[p, q]\}$	correlation matrix
$V = \{V[p, q]\}$	matrix consisting of the set of all eigenvectors of matrix C
$D = \{D[p, q]\}$	diagonal matrix consisting of the set of all eigenvalues of matrix C along its principal diagonal
$W = \{W[p, q]\}$	matrix consisting of the selected eigenvectors of matrix C
$Y = \{Y[m, n]\}$	matrix consisting of the projection of the corresponding vector from matrix X onto the basic vectors contained in matrix W

Step1 Calculate the empirical mean

Find the empirical mean along each of the M dimensions and place the calculated mean values into an empirical mean vector u of dimensions $M \times 1$:

$$u[m] = \frac{1}{N} \sum_{n=1}^N X[m, n]$$

Step 2 Calculate the deviations from the mean

[Miranda 08] found that mean subtraction is an integral part of the solution towards

finding a principal component basis that minimizes the mean square error of approximating the data. We subtract the empirical mean vector \mathbf{u} from each column of the data matrix X and obtain the deviation matrix B with vector \mathbf{h} that is a $1 \times N$ row vector of all 1's:

$$B = X - \mathbf{u}\mathbf{h}$$

Step 3 Calculate the covariance matrix

We work out the $M \times M$ empirical covariance matrix C from matrix B by:

$$C = E[B \otimes B] = E[B \cdot B^*] = \frac{1}{N} B \cdot B^*$$

Where $E[.]$ returned the expectation value, \otimes is the outer product operator and $*$ is the conjugate transpose operator.

Step 4 Find the eigenvectors and eigenvalues of the covariance matrix

We compute the matrix V of eigenvectors which make the covariance matrix C in to a diagonal matrix D consisting of eigenvalues of C :

$$V^{-1}CV = D$$

Matrix D will take the form of an $M \times M$ diagonal matrix:

$$D[p, q] = \begin{cases} \lambda_m, & p = q \\ 0, & p \neq q \end{cases}$$

where λ_m is the m -th eigenvalue of the covariance matrix C . Matrix V contains M column vectors, each of length M , which represent the M eigenvectors of the covariance matrix C . The eigenvalues and eigenvectors are ordered and paired. As a result, the m -th eigenvalue corresponds to the m -th eigenvector.

Step 5 Reorganize the eigenvectors and eigenvalues

We sort the columns of the eigenvector matrix V and eigenvalue matrix D in order of decreasing eigenvalue.

Step 6 Compute the cumulative energy content for each eigenvector

The eigenvalues indicate the distribution of the source data's energy among each of the eigenvectors. The cumulative energy content g for the m -th eigenvector is the sum of the energy content across all of the eigenvectors from 1 through m :

$$g[m] = \sum_{q=1}^m D[p, q]$$

Step 7 Select a subset of the eigenvectors as basis vectors

We build an $M \times L$ matrix W by the first L columns of matrix V :

$$W[p, q] = V[p, q], \quad (1 \leq p \leq M, 1 \leq q \leq L, L < M)$$

We use the vector g as a threshold in choosing an appropriate value for L . The goal is to make L as small as possible while achieving a reasonably high value of g on a percentage basis.

Step 8 Convert the source data to z-scores

We build an $M \times 1$ empirical standard deviation vector s with the square root of each element along the main diagonal of the covariance matrix C :

$$s = \{s[m]\} = \sqrt{C[p, q]} \quad \text{for } p = q$$

Then we will calculate the z-score matrix by:

$$Z[m, n] = \frac{B[m, n]}{(s \cdot h)[m, n]}$$

We know s is an $M \times 1$ vector and h is a $1 \times N$ vector, hence $s \cdot h$ and B are both of dimensions $M \times N$. The every element of Z is the ratio of the corresponding element to the corresponding element of $s \cdot h$.

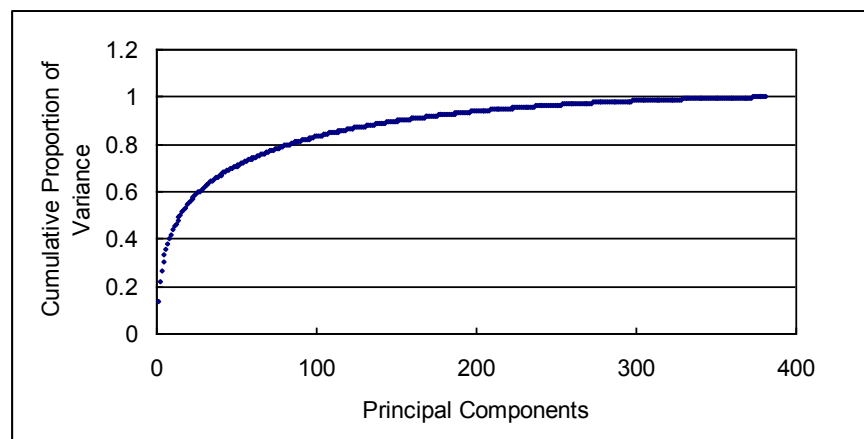
Step 9 Project the z-scores of the data onto the new basis

The projected vectors are the columns of the matrix Y obtained by:

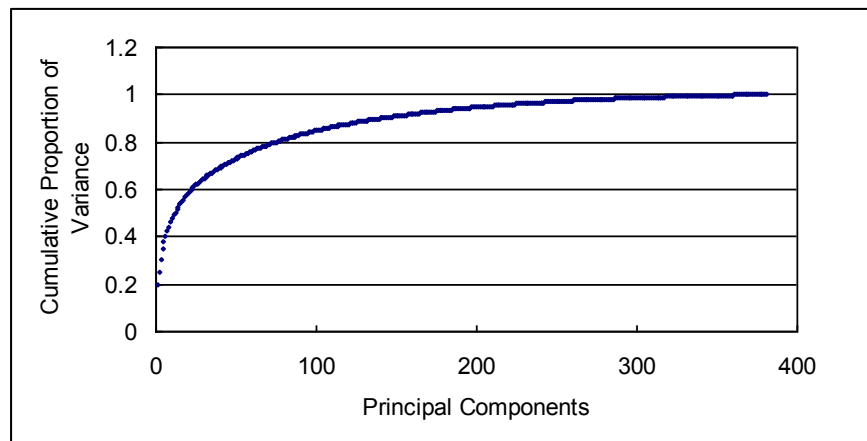
$$Y = W^* \cdot Z$$

The columns of matrix Y represent the Karhunen-Loève transforms (KLT) of the data vectors in the columns of matrix X .

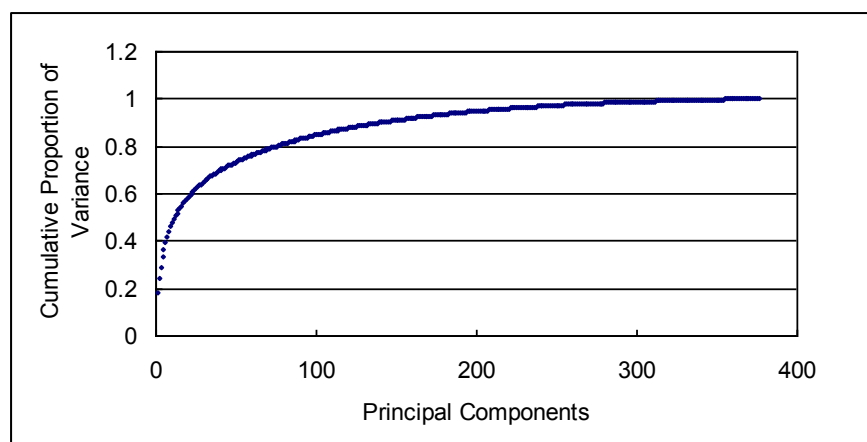
Figure 5.2.2 Cumulative proportion of variance and sorted principal components of all the features of POS bigrams



(Discourse type: advantage/disadvantage. Number of components: 393.)



(Discourse type: advantage/disadvantage. Number of components: 381.)



(Discourse type: advantage/disadvantage. Number of features: 377.)

5.3 Classifier Selection and Comparison

5.3.1 Naïve Bayes classifier

A naïve Bayes classifier is a simple probabilistic classifier by applying Bayes' theorem with strong independence assumptions. It's called "naïve" because of its strong independence assumptions. Depending on the precise nature of the probability model, naïve Bayes classifiers can be trained very efficiently in supervised learning schemes.

In many practical applications, parameter estimation for naïve Bayes models uses the

method of maximum likelihood, which means that one can work with the naïve Bayes model without using any Bayesian methods. Naïve Bayes classifier is very popular for text classification for both documents and web pages, such as pure naïve Bayes classifier of [Lewis 98], [McCallum 98], multinomial naïve Bayes text classifier of [Dumais 98], [Nigam 00] and Possion naïve Bayes proposed by [Kim 03].

In spite of their simple design and apparently over-simplified assumptions, naïve Bayes classifiers often work much better in many complex real-world situations than we might expect. [Lowd 04] found that for a wide range of benchmark datasets, naïve Bayes models have accuracy and learning time comparable to Bayesian networks with context-specific independence, which makes naïve Bayes model a very attractive alternative to Bayesian networks for general probability estimation. Recently, careful analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naïve Bayes classifiers. An advantage of the naïve Bayes classifier is that it requires a small amount of training data to estimate the parameters (such as mean, variances) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The probability model for a classifier is a conditional model $P(C | F_1, F_2, \dots, F_n)$, where F_1, \dots, F_n are feature variables and C is a class variable. We use Bayes' theorem to deduce above conditional probability into:

$$P(C | F_1, F_2, \dots, F_n) = \frac{P(C) \cdot P(F_1, F_2, \dots, F_n | C)}{P(F_1, F_2, \dots, F_n)}$$

The denominator does not depend on C and the values of the feature F_1, \dots, F_n so that it's constant. By repeatedly using the definition of conditional probability we can rewrite the numerator in above expression into:

$$\begin{aligned}
 & P(C) \cdot P(F_1, F_2, \dots, F_n | C) \\
 &= P(C, F_1, F_2, \dots, F_n) \\
 &= P(C) \cdot P(F_1, F_2, \dots, F_n) \\
 &= P(C) \cdot P(F_1 | C) \cdot P(F_1, F_2, \dots, F_n | C, F_1) \\
 &= P(C) \cdot P(F_1 | C) \cdot P(F_2 | C, F_1) \cdot P(F_1, F_2, \dots, F_n | C, F_1, F_2)
 \end{aligned}$$

Now we use the conditional independence assumption: two different features are conditionally independent each other, which mean:

$$P(F_i | C, F_j) = P(F_i | C) \quad \text{for } j \neq i$$

So the probability model can be derived as:

$$P(C, F_1, F_2, \dots, F_n) = P(C) \cdot \prod_{i=1}^n P(F_i | C)$$

Since the posterior probability only depends on the numerator, so the naïve Bayes probability model can be derived as:

$$P(C | F_1, F_2, \dots, F_n) \propto P(C) \cdot \prod_{i=1}^n P(F_i | C)$$

The naïve Bayes classifier combines above model with a decision rule. One commonly used rule is to select the hypothesis that is most probable which is known as the maximum a posteriori or MAP decision rule. The corresponding classifier is the function defined as follows:

$$\text{classifier}(f_1, f_2, \dots, f_n) = \arg \max_c P(C = c) \cdot \prod_{i=1}^n P(F_i = f_i | C = c)$$

Notice that the independence assumption may result in some unanticipated results in the calculation of posteriori probability. In some cases when there is a dependency between observations, the above-mentioned probability may be larger than one.

Despite the fact that the independence assumptions are often inaccurate, the naïve Bayes classifier has several properties that make it astonishingly useful in the real applications. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This can help to alleviate the problems caused by the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. Like all probabilistic classifiers under the maximum a posteriori decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence class probabilities do not have to be estimated very accurately. That's to say, the overall naïve Bayes classifier is robust enough to ignore serious deficiencies in its underlying naive probability model.

5.3.2 Decision tree

In operations research, a decision tree is used to identify the strategy that is most likely to reach a goal. In data mining and machine learning, a decision tree is a predictive model which map from observations about a pattern to conclusions about its target value. In these tree structures, leaves represent classifications and branches represent conjunctions of criteria of features that lead to those classifications. Decision tree has been directly or indirectly used into a lot of applications of computational linguistics, such as word sense disambiguation ([Mooney 96], [Pederson 01]), feature selection for text classification [Berger 06].

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan [Quinlan 93] and it's an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification; hence, C4.5 is often referred to as a statistical classifier.

Based on the concept of information entropy, C4.5 builds decision trees from a set of training data in the same way as ID3. C4.5 uses the fact that each feature of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (difference in entropy) that results from choosing a feature for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision and then the algorithm begin to recur. The pseudocode of C4.5 algorithm is:

Table 5.3.1 Pseudocode of C4.5 algorithm

- | |
|--|
| <ol style="list-style-type: none">1. Check for base cases (whether all patterns belong to one class)2. For each feature f
 Find the normalized information gain from splitting patterns on f3. Let f_best be the feature with the highest normalized information gain4. Create a decision node that splits on f_best5. Recur on the subsets of patterns obtained by splitting on f_best and add those nodes as children of node |
|--|

5.3.3 Logistic regression model

Since our application is to judge whether an IU contains information on a discourse type or not. It can be regarded as a two-class classification so we also try to use logistic regression model [Cox 58] in our experiments. A lot of researches use logistic

regression model in text information retrieval, such as the probabilistic information retrieval model of [Cooper 94], the retrieval models based on bigram indexing ([Chen 01], [Luk 02]) and Bayesian logistic regression algorithm in [Xu 08] used to incorporate relevance feedback information.

In statistics, logistic regression is a model used for prediction of the probability of occurrence of an event by fitting data to a logistic curve of logistic function:

$$f(z) = \frac{1}{1 + e^{-z}}$$

This logistic function is quite useful in various application fields because it can take as an input any value from negative infinity to positive infinity, whereas the output is limited to the values between 0 and 1. The variable z represents the observations to some set of features, while $f(z)$ represents the probability of a particular outcome, given that set of features. The variable z is a measure of the overall contribution of all the features used in the model and is usually defined as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

where β_0 is called intercept and $\beta_1, \beta_2, \dots, \beta_k$ are called regression coefficients. Each of the regression coefficients describes the size of the contribution of the related feature. A positive regression coefficient means that that feature increases the probability of the outcome and a large regression coefficient means that that feature strongly influences the probability of that outcome.

The logits of the unknown binomial probabilities p_i (i.e., the logarithms of the odds) are modeled as a linear function of the feature vector X_i :

$$\ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i}$$

The model has an equivalent formulation:

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i})}}$$

Note that a classifier can be built from this probability model using the maximum a posteriori rule, which means to predict a certain label is above p_i exceeds 1/2:

5.3.4 Support vector machine

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. An SVM regards input data as two sets of vectors in an n-dimensional space and it constructs a hyper-plane to separate the points which present the input data in that space. In the construction of the hyper-plane, SVM maximizes the margin between the two data sets. To calculate the margin, two parallel hyper-planes are constructed on each side of the separating hyper-plane, which are the boundaries between the two data sets. It's obvious that a good separation can be achieved when the hyper-plane has the largest distance to the nearby points of input data of both classes, because generally speaking the larger the margin the better the generalization ability of the classifier.

SVM has very wide application in the field of information retrieval and natural language processing. [Zhang 03] showed that SVM outperforms the classifiers such as Nearest Neighbors, Naive Bayes, Decision Tree and Sparse Network of Winnows for question classification based on two types of features: bag-of-words and bag-of-ngrams. [Joachims 01] developed a theoretical learning model of text classification for SVM and tried to theoretically explain the good performance of SVM

on text classification. Recently, [Raghavan 07] propose a solution for a better feature selection for SVM based text classifier.

Suppose we have some given data and each of them has a label from two classes, and the goal of classification is to decide which class a new data point will belong to. When an SVM is used for classification, a data point is viewed as a p -dimensional vector, and we want to know the feasibility to separate such points with a $p-1$ -dimensional hyper-plane. This is called a linear classifier. There are many hyper-planes that might classify the data. However, we are especially interested in finding out the existence of the maximum separation margin between the two classes. We will select the hyper-plane so that the distance from the hyper-plane to the nearest data point is maximized. Hence, if such a hyper-plane exists, it is undoubtedly of interest and is known as the maximum-margin hyper-plane and such a linear classifier is known as a maximum margin classifier.

Given a set of training data D and D has the form:

$$D = \{(x_i, L(x_i)) \mid x_i \in R^p, L(x_i) \in \{1, -1\}\}$$

where R^p denotes p -dimensional real vector, $L(x_i)$ is the label of x_i is 1 or -1, indicating the class to which the point x_i belongs. We want to find the maximum-margin hyper-plane which divides the points having label 1 from those having label -1. We know that any hyper-plane can be written as the set of points x that can satisfy:

$$w \cdot x - b = 0$$

Vector w is perpendicular to the hyper-plane and $b/\|w\|$ determines the offset of above hyper-plane from the origin along vector w . Since we need to select w and b to

maximize the margin and also enable the hyper-plane to separate the training data.

These hyper-planes can be described by:

$$w \cdot x - b = 1 \quad \text{and} \quad w \cdot x - b = -1$$

If the training data are linearly separable in the p -dimensional space, we can find two hyper-planes such that there are no data points between them and we can try to maximize the distance between the two hyper-planes. The distance between them is $2/\|w\|$, so we need to minimize $\|w\|$. In order to make sure there are no points located in the margin, the points having label 1 should satisfy: $w \cdot x_i - b \geq 1$ and the points having label -1 should satisfy: $w \cdot x_i - b \leq -1$. After bringing the labels, we can merge above two conditions into: $L(x_i) \cdot (w \cdot x_i - b) \geq 1$. The optimization problem can be described as:

To choose w, b to minimize $\|w\|$ subject to $L(x_i) \cdot (w \cdot x_i - b) \geq 1$.

The original optimal hyper-plane algorithm proposed by Vladimir Vapnik in 1963 was a linear classifier. However, [Boser 92] suggested a way to create non-linear classifiers by applying the kernel trick (originally proposed by [Aizerman 64]) to maximum-margin hyper-planes. According to Mercer's theorem, any continuous, symmetric, positive semi-definite kernel function $K(x, y)$ can be expressed as a dot product in a high-dimensional space. Kernel trick is a method of using a linear classifier algorithm to solve a non-linear problem by mapping the original non-linear observations into a higher-dimensional space, where the linear classifier is subsequently used; this makes a linear classification in the new space equivalent to non-linear classification in the original space.

The resulting algorithm is formally similar with the linear ones, except that every dot product is substituted by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyper-plane in the transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyper-plane in the high-dimensional feature space it may be non-linear in the original input space. Some common kernels are:

- Polynomial (homogeneous): $k(x, x') = (x, x')^d$
- Polynomial (inhomogeneous): $k(x, x') = (x, x' + 1)^d$
- Radial Basis Function: $k(x, x') = \exp(-\gamma \|x - x'\|^2), \gamma > 0$
- Sigmoid: $k(x, x') = \tanh(\kappa x \cdot x' + c)$

5.4 Experimental Results

Figure 5.4.1 illustrates the architecture of our classification system. Since relevant documents are much less than irrelevant ones for any query, we use all the information of relevant documents by formulating all the IUs of relevant documents as positive instances (patterns). We also select the top-ranked irrelevant documents of retrieved list to obtain negative patterns. When we build these patterns, an IU is a pattern and each linguistic sequence (e.g. POS tag bigram) is a feature. The feature value is the frequency of the corresponding linguistic sequence. First we preprocess the patterns by removing the useless features that occur extremely frequently or infrequently. According to Figure 5.2.1, we set the threshold of standard deviation δ as 0.1, 0.2 and

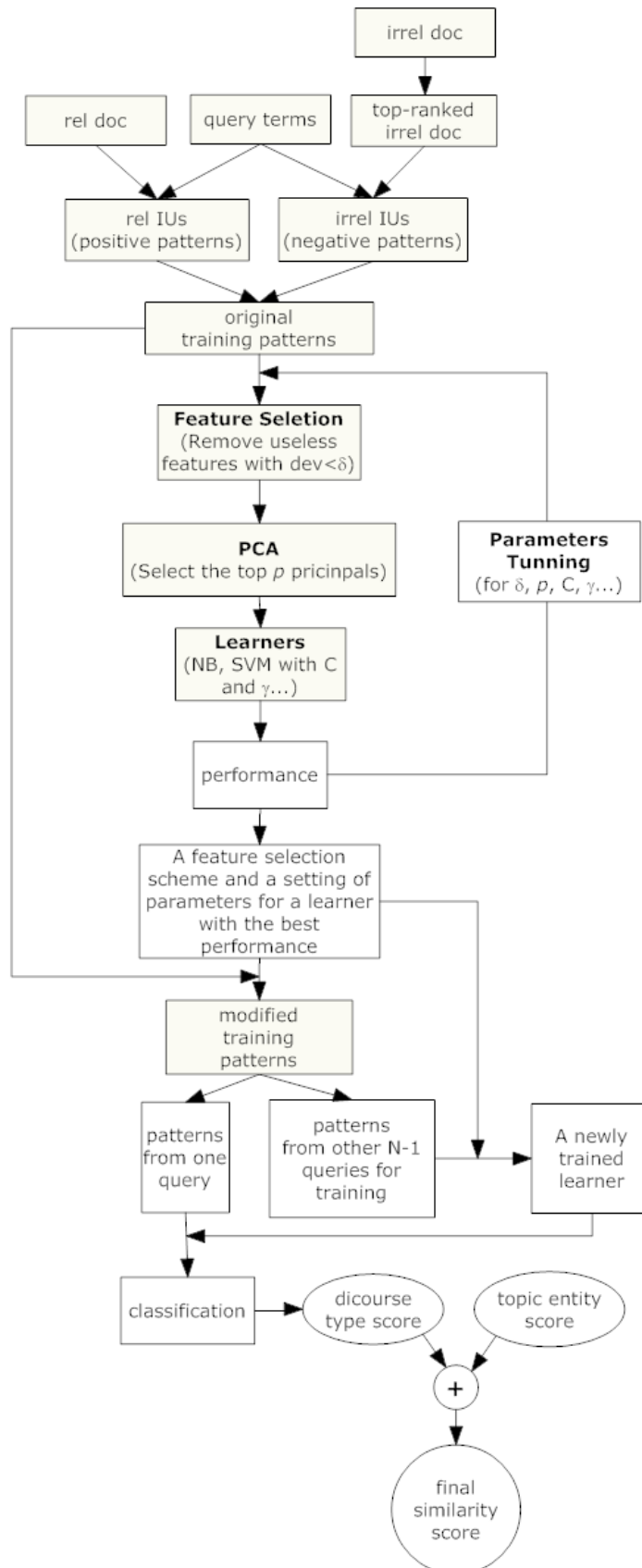
0.5. The number of the features greatly decreases and it makes the later Principal Component Analysis cost-efficient. Since the principal components are sorted by the eigenvalues, we select top p of the components to further simplify the later training and testing for the different classifiers. Some classifiers such as SVM have some parameters and these parameters together with above δ and p can be determined by the cross validation on the patterns. A K-fold cross validation is used and K is set to 5 and shuffled sampling on the pattern for cross validation is used. After the “optimal” parameters are determined for a specific classifier and we will use this classifier to predict an outcome or probability of how an IU is related with the discourse type.

We use accuracy to evaluation the performance of a classifier and accuracy is defined as the ratio of the number of the patterns that to be predicted correctly to the total number of patterns. Since we are using 5-fold cross validation, each pattern is tested once so the numerator of the following formula is the summation of the five numbers; each corresponds to the number of correctly predicted patterns. The denominator is the total number of the patterns.

$$accuracy = \frac{\#(true+, pred+) + \#(true-, pred-)}{\#(true+, pred+) + \#(true+, pred-) + \#(true-, pred+) + \#(true-, pred-)}$$

In this section, we are using eight topics of advantage/disadvantage as examples because this discourse type has more selected IUs than the other two. We present the testing results of Naïve Bayer with different standard deviation thresholds (δ) and number of principal components (p) in Table 5.4.2. “# of f” denotes the number of features after removal. For example, when we set δ to 0.5, we can get 30 features.

Figure 5.4.1 Illustration of the design of our classifiers



After we perform PCA, we select top 5, 10, 20, 30 principal components to do the 5-fold cross validation. From the results, we find that the different δ 's and p 's do not results in very different results. Obviously, bigger p will generally lead to better results. From the results, we conclude that the best performance occur when δ is set to 0.2.

Table 5.4.2 Cross validation results in accuracy (%) of Naïve Bayer with different number of principal components and standard deviation removal threshold

		p (# of principal components) =								
	# of f	5	10	20	30	50	100	150	200	all
$\delta=0.5$	30	59.81	60.55	60.73	60.90	NA	NA	NA	NA	60.90
$\delta=0.2$	128	60.00	61.01	61.00	60.34	61.21	62.71	NA	NA	62.75
$\delta=0.1$	217	59.97	61.11	60.82	60.48	61.38	62.38	62.57	62.12	62.10

(Discourse type: advantage/disadvantage. Linguistic sequence: POS tag bigram. Number of features: 1044. Number of patterns: 11193)

Table 5.4.3 Cross validation results in accuracy (%) of decision tree C4.5 with different number of principal components and standard deviation removal threshold

		p (# of principal components) =								
	# of f	5	10	20	30	50	100	150	200	all
$\delta=0.5$	30	61.65	62.40	62.51	61.28	NA	NA	NA	NA	61.28
$\delta=0.2$	128	62.34	61.74	62.47	61.48	62.75	61.99	NA	NA	62.13
$\delta=0.1$	217	61.99	62.30	62.20	62.20	62.67	62.39	62.52	62.92	63.05

(Discourse type: advantage/disadvantage. Linguistic sequence: POS tag bigram. Number of features: 1044. Number of patterns: 11193)

In Table 5.4.3, we present the results from the same setting with Table 5.4.2 by using decision tree C4.5 as classifier. In C4.5 algorithm, we use information gain as splitting criterion and the minimal size of node allowing for split is set to 4. The minimal size of all leaf nodes is set to 2 and maximal tree depth is set to 10 and the confidence level for pessimistic error calculation of pruning is set to 0.25. From the results, we can also find that different δ 's and p 's do not results in very different results.

Table 5.4.4 Cross validation results in accuracy (%) of different classifiers

$\delta=0.5$	p (# of principal components) =			
	5	10	20	30
Naïve Bayes	59.81	60.55	60.73	60.90
Decision Tree	61.65	62.40	62.51	61.28
Logistic Regression	60.47	61.69	63.08	63.47
SVM (RBF kernel)	62.47	64.44	68.71	71.79

(Discourse type: advantage/disadvantage. Linguistic sequence: POS tag bigram. Number of features: 1044. Number of patterns: 11193)

Table 5.4.4 shows the performance in accuracy of different classifiers. In addition to naïve Bayes and decision tree, we also use logistic regression model and support vector machine (SVM). We are doing 5-fold cross validation for the four classifiers within the same group of patterns. We can see that the performance of SVM is better than the other three classifiers which have very close results. (Note that we are using LIBSVM [Chang 01] as SVM classifier. The SVM in Table 5.4.4 is using radial basis function kernel with parameter $C=0$ and $\gamma=0.1$)

There are four basic kernel functions for SVM: linear, polynomial, radial basis function (RBF) and sigmoid. We are using RBF kernel function in our experiments because, according to [Chang 01], RBF kernel nonlinearly maps samples into a higher dimensional space so it can handle the case when the relation between pattern labels and features is nonlinear. Furthermore, the linear kernel is a special case of RBF as [Keerthi 03] shows that the linear kernel with a penalty parameter has the same performance as the RBF kernel with some parameters (C, γ). Additionally, the sigmoid kernel behaves like RBF for certain parameters [Lin 03]. The second reason is the number of hyper-parameters which determine complexity of model selection. The polynomial kernel has more hyper-parameters than the RBF kernel.

Table 5.4.5 Results in accuracy (%) of SVM using RBF kernel with different parameters

$\delta=0.5$	$\gamma =$				
C =	0.03125	0.125	0.5	2	8
1	63.08%	65.98%	73.31%	93.82%	99.61%
2	64.40%	66.94%	77.10%	97.79%	99.71%
8	65.42%	69.56%	85.86%	99.61%	99.73%
32	66.74%	73.81%	94.09%	99.71%	99.74%
128	68.55%	79.20%	98.70%	99.72%	99.74%

(Discourse type: advantage/disadvantage. Linguistic sequence: POS tag bigram. Number of features: 1044. Number of patterns: 11193)

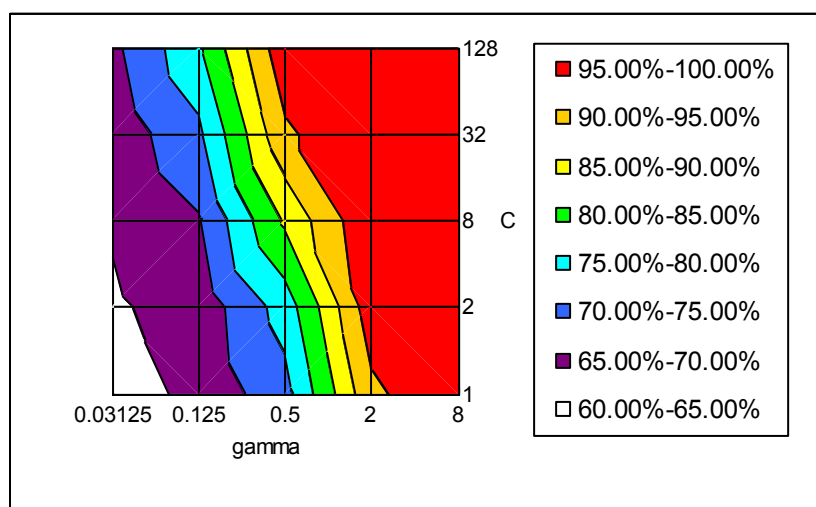


Table 5.4.6 Cross validation results in accuracy (%) of SVM using RBF kernel with C=100

$\delta=0.2, p=100$	$\gamma =$						
	0.005	0.01	0.02	0.05	0.07	0.1	0.2
Accuracy	65.32	83.32	83.56	81.80	80.93	78.25	68.95

(Discourse type: advantage/disadvantage. Linguistic sequence: POS tag bigram. Number of features: 1044. Number of patterns: 11193)

We perform a “coarse grid search” to find the best parameters (C, γ) for SVM with RBF kernel for our available patterns, which is suggested by [Chang 01]. An effective way is to use the same data for both training and testing. Note that the results in Table 5.4.5 are not obtained by cross-validation. We try parameter C from $2^0, 2^1, 2^3, 2^5$ and 2^7 and parameter γ from $2^3, 2^1, 2^{-1}, 2^{-3}$ and 2^{-5} . The features are selected by removing all features with δ less than 0.5 and perform PCA on all the patterns. The results are

shown in Table 5.4.5.

In order to evaluate and confirm our parameter search methods based on “testing-on-training-data”, we also do 5-fold cross-validation on a larger scale of data. We set δ to 0.2 so that we can use more features (127 features). Hence the selection of γ is somewhat different from $\delta=0.5$. We set C of RBF function to 100 and Table 5.4.6 shows some the best results among different values of γ .

Table 5.4.7 Re-ranking performance in MAP using SVM with RBF kernel based on POS tag bigrams

C=100		$\gamma =$						
Disc Type	baseline	0.005	0.01	0.02	0.05	0.07	0.1	0.2
Adv/dis	0.209	0.214	0.230	0.227	0.224*	0.225	0.221	0.227
Reason	0.224	0.230	0.246	0.244^	0.243	0.241	0.236	0.242
Impact	0.229	0.233	0.251	0.246#	0.245	0.247	0.241	0.248
Mean	0.222	0.227	0.243	0.240	0.237	0.238	0.234	0.240

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

We combine the predication results of SVM with the original similarity score of BM 11 and re-rank the retrieved list. In order to make comparison with the re-ranking results form other methods, we use the re-ranking scoring formula with the same setting:

$$rel(doc, q) = \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} \cdot K[SVM(IU_i)] \right\}$$

where S_0 is original similarity score of the i -th IU. $SVM(IU_i)$ is the SVM’s prediction value on the i -th IU. $K[.]$ is a function defined by:

$$K(c) = \begin{cases} k, & c = 1 \\ 1, & c = -1 \end{cases}$$

Obviously, when the prediction value is -1 (irrelevant), the function return one and when the prediction value is 1 (relevant), the function returns k . In our experiments, k is set to 5. The original similarity score is combined with score of SVM by multiplication. For each query, the SVM is trained by the IUs of relevant documents and selected top-ranked irrelevant documents of other seven queries. The SVM is using RBF kernel with $C=100$. According to the results in Table 5.4.6, we try different values for γ . We present the re-ranking performance in Table 5.4.7 for all the three discourse types. We find the results with highest mean MAP occur at $\gamma=0.01$ but the most significantly improve results occur at $\gamma=0.02$ and 0.05 .

From Chapter 4 we know that “pw” type is the best sequences among all the sequences types. We perform experiments on “pw” type with SVM. Compared with POS tag bigram, the distinct number of “pw” type sequences is much bigger, which will greatly increase the complexity of the learning procedure. We reduce the number of features by the following two ways. First, we remove the “pw” type sequences containing the words that rarely occur. Second, we merge some POS tags into one general tag according to Table 5.4.8. We group different types of nouns, different inflected verbs and different punctuations together.

We use the same parameter setting as POS tag bigrams because we find that the “pw” type sequences have very similar performance in selecting parameters with POS tag bigrams after reducing the features. We present the results in Table 5.4.9 and the performance of using “pw” type sequences is consistently better than the baseline

when parameter γ is set to 0.02. By comparing the results of POS tag bigrams and “pw” type sequences, we find that the “pw” results are a little bit higher but the differences are not very big.

Table 5.4.8 POS tag merging

Previous POS tags	General POS tag
NN, NNP, NNPS	N
VB, VBD, VBG, VBN, VBP, VBZ	V
\$, “, ”, (,), ,, --, ., :	PUNC

Table 5.4.9 Re-ranking performance in MAP using SVM with RBF kernel based on “pw” type sequences

C=100		$\gamma =$						
Disc Type	baseline	0.005	0.01	0.02	0.05	0.07	0.1	0.2
Adv/dis	0.209	0.235	0.231	0.229*	0.224	0.223	0.233	0.235
Reason	0.224	0.248	0.245	0.243^	0.247	0.237	0.245	0.248
Impact	0.229	0.253	0.247	0.245#	0.246#	0.244	0.248	0.253
Mean	0.222	0.246	0.242	0.240	0.240	0.236	0.243	0.246

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

In conclusion, we formulate the problem of determining whether an IU contains the discourse type information into a pattern recognition application. An IU is modeled as a vector and a linguistic sequence of a certain type is a feature. Feature selection is performed on the patterns by removing the features with standard deviation smaller than a threshold. Principal Component Analysis is also used to enhance the features. Experimental results shows that SVM with RBF kernel is better than Naïve Bayes, decision tree C4.5 and logistic regression model. We put the SVM’s prediction value

into re-ranking and the original retrieval performance can be generally improved.

Summary

In this Chapter, we formulate the discourse type based retrieval into a pattern recognition problem. Then we use some selected linguistic sequences as features and use different classifiers to judge whether an IU is relevant or not. According to the experiments, we find that support vector machine (SVM) with RBF kernel is the best classifier for our problem and it can generate significantly improved results with the appropriate parameters setting. Compared with our computation models in Chapter 4, the SVM is not as good as our models due to the apparent difficulties of the problem: the number of features is much larger than the number of instances. In order to ensure the terms in an IU are highly related to the centre topic entity term, the number of terms in an IU cannot be very large so that only a few features have meaningful values for all the instances. This special situation leads to the poor performance of the traditional classifiers that is why we propose our computation models in Chapter 4.

CHAPTER 6

CONCLUSIONS

This thesis creatively puts forward the concept of discourse type to more accurately and completely describe an information need. In practical applications and experiments, an information need always is given by a query. How to recognize the discourse type of a query is not the focus of this thesis. This thesis provides the solutions for retrieval after a discourse type of a query is known. By manually examined all the TREC Robust Track topics, we classify these topics according to their discourse type(s). We choose the discourse types with poor retrieval performance as examples in our study.

We extract the context around topic entity terms in the documents to form a fixed-size text passage which are called information unit (IU). Then we simplify our problem into determining whether an IU is relevant to a given topic. There are two factors to determine the relevance of an IU: relevance with the topic entity and relevance with the discourse type.

We propose two types of IU similarity models to compute the relevance (measured by similarity) of IU with a topic entity. These two models are fuzzy model and graph-based model and they can be a baseline of not using discourse type information for IU-based retrieval.

The fuzzy models with different term weighting schemes (constant term

weighting, feature-based term weighting and fuzzy set term weighting) have different performances. Re-ranking based on fuzzy models with three types of term weighting schemes can all enhance the original retrieval performance. Among these three types of term weighting, constant term weighting contributes the best results with **SUMtf**. Different distance measuring function can affect the results of feature-based term weighing if distance factor is considered. The best result of **SP** (feature-based term weighing based on distance factor only) is very close to **SUMtf** and this result is based on a non-convex exponential function. The best results of re-ranking based on fuzzy set term weighting are better than the original results but worse than the two results mentioned above.

The basic unit of graph-based models is a pair of terms linked by an edge. To select all edges is better than other edge selection schemes that select part of edges. However, the other four edge selection schemes are a little worse than selecting all edges, which can reflect the importance of some special edges in an IU graph to the relevance of an IU. Therefore we think that edge selection schemes that select part of edges according to some criteria are still potential, especially when IU size is big. As for the calculation of edge relevant evidence, “*ord*”, which consider the order of the linked two terms, produces the best results for all of the five edge selection schemes. It shows that term order is more important than the distance between two terms or term specialty in quantifying the relevance of an edge that links two terms.

In order to discover the phrases or text patterns that can express a discourse type, we use word sequences (bigram, trigram, 4-gram), POS-tag sequences (bigram, trigram, 4-gram, 5-gram) and word-POS tag sequences (bigram, trigram) as features to

match the text in the IUs of the relevant documents of the topics with some specific discourse types. Some features are proposed in order to select the representative sequences that are prevalently used for a specific discourse type. These features can help to discover the discourse type related sequences with very small occurrence frequencies.

Among the word sequences of different lengths, word bigram is the best in improving the retrieval by re-ranking and therefore to increase the length word sequence does not help. Based on the results of the retrospective result, the feature DFR m2, i.e. query DFR (document frequency ratio) aggregated by the m2 method (macro average), is the best feature to provide the sorting standards for word sequences that can consistently yield good results. Also, among the query frequency features, QF n2, the ratio of relevant query frequency to the irrelevant query frequency, is also a good feature to sort word sequences, which is comparable with DF m2.

Among the POS tag sequences of different lengths, POS tag 4-gram is the best POS tag sequence in improving the retrieval by re-ranking. DFR m2 feature, which relies on query DFR (document frequency ratio) measures with the m2 aggregation method (macro average), is the best measure to sort POS tag sequences for re-ranking. Also, among the query frequency features, QF n2, the ratio of relevant query frequency to the irrelevant query frequency, is also a good function to sort word sequences.

Among word-POS tag sequences (bigrams and trigrams), word POS tag bigram with type “pw” is the best in improving the retrieval by re-ranking and second best one is word-POS tag trigram with type “ppw”. The DFR m2 feature, which relies on query IUFR (IU frequency ratio) measures with the m2 aggregation method (macro average),

is the best measure to sort word-POS tag sequences for re-ranking.

If we make a comparison among all types of sequences, word-POS tag sequences are generally better than POS-tag sequences, which are better than word sequences. POS 4-gram and 5-gram are comparable with “pw” and “ppw” type of word-POS tag sequences. We investigate the possibility of using the combination of different types of the linguistic sequences. The consequence is that *ceteris paribus* to only use “pw” type sequences always outperforms the runs that use other types of sequences. It is because the “pw” type sequences have incomparable detecting ability and other types cannot make a supplementary support. We also compare our features with popular term measure such as Robertson’s w4 and the best ones of our features (e.g. QF n2) is better than w4 in that our features better reflect the distribution of an infrequent sequence in the relevant IUs of different queries.

We propose two principles for mining discourse type related sequences. The first principle is that the sequences must occur in more documents. The second principle is that the sequences must occur differently in relevant and irrelevant documents. We use normalized Zipf’s curve to evaluate different types of sequences.

Normalized Zipf’s curve can indicate the overall distribution of the sequences with the same type based on their likelihood to occur in more documents. The query level measures and query set level features intend to find the difference of the occurrence of a sequence in relevant and irrelevant documents, which is to quantify the requirement of the second principle. The normalized Zipf’s curve is to investigate which sequence type has a large percentage sequences that are likely to occur in more documents, which is to quantify the requirement of the first principle. Our normalized Zipf’s curve

can reflect the quality of sequence for detecting and presenting discourse types. The “pw” type word-POS tag sequence is the best sequence we’ve found based on our study. Closer to the normalized Zipf’s curve of “pw” type sequence, the better respective and actual re-ranking results a sequence type can produce.

We also formulate the problem of determining whether an IU contains the discourse type information into a pattern recognition application. An IU is modeled as a vector and a linguistic sequence of a certain type is a feature. We perform feature selection by removing the features with standard deviation smaller than a threshold. Principal Component Analysis is also used to enhance the features. Experimental results shows that SVM with RBF kernel is better than Naïve Bayes, decision tree C4.5 and logistic regression model. We put the SVM’s prediction value into re-ranking and the original retrieval performance can be generally improved.

APPENDIX

A1. Retrospective retrieval experiments based on word sequence

We did some retrospective experiments to test the features of word sequences and these experiments are performed on the word sequences extracted from the IUs of the relevant documents. In the last subsections, for each word sequence and each query, we propose the distribution measures which include relevant document frequency, irrelevant document frequency, relevant IU frequency, irrelevant IU frequency and word sequence frequency for relevant/irrelevant documents/IU and ratios of above measures. In order to make clear that these features are for a word sequence and a given query, we add the word “query” before the basic measure name, such as “query DFR”. Also, we introduce the ways of deriving these “query measures” based on the basic measures. In this subsection, we further derive the query measures into query set features, which indicate the distribution of a word sequence in a set of queries with the same discourse type. In query set level, we call “feature” because these features are actually used in the sorting of the word sequences.

After we obtain a query set feature of a word sequence for a set of queries with the same discourse type, we will rank all the extracted word sequences according to this feature. We have various query set features and these word sequences are ranked in descending order according to each of the features. For each query set feature, we select a certain number of the word sequences from the ranked list as discourse type

terms. We use the following formula A.1 to compute the re-ranked score for the retrieved documents:

$$\begin{aligned}
S' &= \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} + sim[IU_i(doc), D] \right\} \\
&= \sum_{i=1}^{|doc|_{IU}} \left\{ \frac{S_0}{|doc|_{IU}} + \sum_{dt_j \in D} tf[IU_i(doc), dt_j] \right\} \quad (A.1)
\end{aligned}$$

In above formula, S' is the new score for re-ranking a document. S_0 is the original similarity score obtained by our baseline retrieval model and the original retrieved list is sorted according to S_0 . $|doc|_{IU}$ is the number of IUs in the document doc which is equal to the number of the topic entity terms in doc . $IU_i(doc)$ is the i -th IU of document doc . D is a set of discourse terms and in the retrospective experiments these terms are top ranked word sequences in the sorted list described above. $sim[]$ denotes the similarity between an IU and discourse term set D . A simple way to derive $sim[]$ is to use the number of the word sequence of D that occur in the IU, viz. $tf(IU, D)$, which is the total number of the word sequence in D occurring in the i -th IU. We use summation to combine this number with the item before it because it is the simplest way to assure that if any discourse related word sequence occurs in any IU, the total score is increased and if no discourse related word sequence occurs, the original score remains the same. We assign the original score $S_0/|doc|_{IU}$ to each IU in a document in order to quantify the contribution of an IU to the whole document, since the number of IUs in the documents are quite different. The theory behind the concept of IU is that the context terms, the terms around the topic entity term, is more related with and provides more important information for the term at the centre of IU than the non-context terms (the terms that occur far way from the term at the centre). Similarly, the influence and

support of a discourse type related word sequence is only limited inside each IU. Hence, we compute the scores for each of the IUs based on the occurrence of discourse related word sequence and then aggregate the scores of all the IUs together as a new document scores.

We propose three ways (they are m1, m2 and m3) to generate query set feature by aggregating the three types of query measures (they are query DFR, query IUFR and query WFR). For query DFR, which is ratio of the number of relevant documents that contain the word sequence to the number of the irrelevant documents in the retrieved list of a query, query set DFR feature m1 is the micro average of query DFR, we have the following function to compute query set DFR:

$$DFR_{m1}(S, ws) = \frac{1}{T} \sum_{i=1}^T DFR(q_i, ws) = \frac{1}{T} \sum_{i=1}^T \frac{RDF(q_i, ws)}{IDF(q_i, ws) + \alpha}$$

Query set DFR m2 is the macro average of query DFR and we use the following function to compute query set DFR:

$$DFR_{m2}(S, ws) = \frac{\sum_{i=1}^T RDF(q_i, ws)}{\sum_{i=1}^T IDF(q_i, ws)}$$

Query set DFR m3 is the macro average to compute the normalized query DFR. The formula to compute this feature is:

$$DFR_{m3}(S, ws) = \frac{\sum_{i=1}^T [RDF(q_i, ws) / R(q_i)]}{\sum_{i=1}^T [IDF(q_i, ws) / I(q_i)]}$$

The query measure aggregation methods to generate DFR m1, m2, m3 are also adopted for IUFR and WSF. For query IUFR which is ratio of the number of relevant IUs that contain the word sequence to the number of the irrelevant IUs of the

documents in the retrieved list of a query, query set IUFR feature m1 is the summation of query IUFRs of all the queries, see the below formula. Since it's summation of the ratios of all the queries, it's just like the micro average of the ratios.

$$IUFR_{m1}(S, ws) = \sum_{i=1}^T IUFR(q_i, ws) = \sum_{i=1}^T \frac{RIUF(q_i, ws)}{IIUF(q_i, ws)}$$

Query set IUFR m2 feature is the ratio of the total number of the relevant IUs that contains the word sequence to the total number of the irrelevant IUs, see the below formula. Since it's the ratio of the two summations, it's just like the macro average of the query IU frequency ratios.

$$IUFR_{m2}(S, ws) = \frac{\sum_{i=1}^T RIUF(q_i, ws)}{\sum_{i=1}^T IIUF(q_i, ws)}$$

Query set IUFR m3 feature is based on the normalized query RIUF and IIUF, see the below formula. The query RIUF is normalized by considering the number of IUs in all the relevant documents of query q_i . The query IIUF is normalized in the same way. Feature m3 is the ratio of the summations of the two normalized query IU frequencies. It's just like the macro average of the ratios.

$$IUFR_{m3}(S, ws) = \frac{\sum_{q_i} [RIUF(q_i, ws) / \sum_{m=1}^{R(q_i)} |rd_m|_{IU}]}{\sum_{q_i} [IIUF(q_i, ws) / \sum_{n=1}^{I(q_i)} |id_n|_{IU}]}$$

Query WSFR is the ratio of the number a word sequence in the IUs of all the relevant documents to the number of the word sequence in the IUs of the selected

irrelevant documents in the retrieved list of a query. Query set WSFR feature m1 is the summation of the query WSFRs of the queries that have the same discourse type. Since query WSFR is the ratio of the query WSF of the relevant documents to query WSF of the selected irrelevant documents, feature m1 can be regarded as the micro average of the ratios. The following formula is to compute query set WSFR m1 feature. For the reason of brevity, we don't repeatedly derive the "*Rfreq(.)*" and "*Ifreq(.)*", which are explained in details in last subsections.

$$WSFR_{m1}(S, ws) = \sum_i WSFR(q_i, ws) = \sum_i \frac{Rfreq(q_i, ws)}{Ifreq(q_i, ws)}$$

We use the following formula to compute query set WSFR feature m2, which is the ratio of the total frequency of *ws* in the IUs of the relevant documents in the retrieved lists of all the queries to the total frequency of *ws* in the IUs of the selected irrelevant documents. We can regard feature m2 as the macro average of the frequency ratios.

$$WSFR_{m2}(S, ws) = \frac{\sum_i Rfreq(q_i, ws)}{\sum_i Ifreq(q_i, ws)}$$

Query set WSFR feature m3 is based on the normalized query WSFRs, which consider the total number of word sequences in the IUs of the relevant and irrelevant document. In the formula, the number of IUs is used instead of the number of word sequences because IU has fixed size. The following formula is to compute feature m3:

$$WSFR_{m3}(S, ws) = \frac{\sum_{q_i} [Rfreq(q_i, ws) / \sum_{m=1}^{R(q_i)} |rd_m|_{IU}]}{\sum_{q_i} [Ifreq(q_i, ws) / \sum_{n=1}^{I(q_i)} |id_n|_{IU}]}$$

So we have proposed three ways to aggregate the measures of query DFR into

feature DFR m1, m2 and m3. We also aggregate query IUFR and query WSFR. As a result, there are totally nine methods to score a word sequence. In addition, we propose three features based on the query frequency, which indicates how many queries in a query set meet a given condition. These query frequency based features are quite different with the nine methods but they are direct and easy to compute. Experiences show that they are comparable measures with the above nine methods.

The first query frequency based feature is relevant query frequency, denoted by QF n1, which is the number of queries that have a given word sequence in the IUs of relevant documents no matter how many times it occurs. This feature is good at evaluating the very specific word sequences. It can indicate the distribution of a word sequence in the relevant sets of a set of queries. We know that according to the Zipf's law only a few words are used very often, many or most are used rarely. The word sequences also occur in the same way. It's possible for a word sequence to occur once in an IU of a relevant document of one query and it also occurs in an IU of a relevant document of another query. QF n1 feature can help to recognize this type of word sequences though its word sequence frequency and IU frequency are very small.

Irrelevant query frequency is the counterpart of relevant query frequency, which is the number of queries that have a given word sequence in the IUs of the selected irrelevant documents. The ratio of relevant query frequency to the irrelevant query frequency, denoted by QF n2, is another feature to indicate the difference of a word sequence's distribution in relevant document and irrelevant documents. In order to avoid zero division, a constant 0.1 is added for the denominator.

Given the RDF (relevant document frequency) and IDF (Irrelevant document

frequency) of a word sequence and a query, if RDF is bigger than IDF, we can conclude that the word sequence is more likely to occur in the relevant documents than irrelevant documents, since the relevant set and selected irrelevant set have the same number of documents. We also propose feature QF n3 which is the number of queries that meet the condition that RDF of the word sequence is bigger than its IFD. Let take word bigram “*is more*” as an example, “*is more*” occurs in the IU of one relevant document of topic 308, three relevant documents of topic 605, see column “Query RDF” in Table A.1.1. “Query IDF” is the number of irrelevant documents that contain “*be more*” in the IUs. So query set RDF of “*is more*” is 24 and query set IDF is 5. Since there are six queries that contain “*is more*” in the IUs of relevant documents, so QF n1 measure of “*is more*” is equal to 6. Since there are three queries that contain “*is more*” in the IUs of irrelevant documents, QF n2 measure of “*is more*” is equal to 1.94 (6/3+0.1). There are six queries that have RDF value bigger than IDF, so the QF n3 measure of “*is more*” is 6.

Table A.1.1 Query RDF, query IDF of word bigram “be more” for each topic

Topic ID	number of documents in relevant set	Query RDF	Query IDF	RDF>IDF?
308	4	1	0	Yes
605	63	3	1	Yes
608	25	1	0	Yes
624	18	0	0	No
637	22	1	0	Yes
654	48	10	0	Yes
690	6	0	1	No
699	66	8	3	Yes
Total	252	24	5	6

Table A.1.2 RDF, IDF measures and QF n3 feature of some examples of word bigrams and trigrams

Discourse Type: Advantage/disadvantage			
Word Bigrams	Query set RDF	Query set IDF	QF n3
not be	31	13	7
the most	39	21	6
is more	24	5	6
at least	29	13	6
the number	32	19	5
should not	11	0	5
the biggest	12	4	5
quality of	25	9	4
majority of	12	3	4
are increasingly	4	0	4
compared with	19	3	3
the top	36	4	3
Word Trigrams			
should not be	9	3	5
will not be	8	5	4
to increase the	7	0	4
it is not	12	6	5
it would not	5	0	4
it was not	4	0	4
than the average	4	0	3

In Table A.1.2, we present the query set RDF, IDF measures and QF n3 feature of some word sequences. We can see that query set RDF of “*than the average*” is 4 and its query set IDF is zero. Based on the re-ranking formula A.1 proposed at the beginning of this subsection, we can conclude that this word trigram can definitely improve the retrieval performance of this set of queries because this word trigram never occurs in any IUs of the irrelevant documents. The value of query set RDF and word sequence frequency of this word trigram are not large due to the number of

relevant documents and the limited size of an IU. We can still discover this word trigram by using query frequency features such as QF m2 or QF m3. This case is to show the purpose of putting forward the query frequency measures.

After the word sequences extracted from the IUs of the relevant documents are sorted according to the one of above features, we select a certain number of top ranked word sequences as discourse type word sequences, which compose D in above re-ranking formula A.1. For each sorted word sequence list, we respectively select the top 50, 100, 200, 500, 1000 2000 and 5000 word sequences as D to re-rank the original retrieved list. After we re-rank the original retrieved list in accordance with the formula proposed at the beginning of this subsection, we obtain the new MAPs and we can compare it with the original one and perform significant tests.

Table A.1.3 Retrospective re-ranking retrieval performance based on word bigrams (Part I)

Discourse Type: Advantage/disadvantage								
baseline		Top N of discourse type related word bigrams, N=						
=0.2086		50	100	200	500	1000	2000	5000
DFR	m1	0.1205	0.1790	0.2363	0.2424	0.2756	0.2998	0.3126*
	m2	0.1414	0.1706	0.2291	0.2916	0.3986*	0.4505*	0.5098*
	m3	0.1711	0.2271	0.2923	0.3111	0.3151	0.3276	0.3144*
IUFR	m1	0.1125	0.1493	0.2085	0.2711	0.2743	0.2939	0.3030*
	m2	0.0895	0.1938	0.2391	0.3159	0.4078*	0.4666*	0.5248*
	m3	0.2360	0.2626	0.2058	0.2189	0.2285	0.2409*	0.2569*
WFR	m1	0.1296	0.1738	0.1971	0.2541	0.2746	0.2865	0.3030*
	m2	0.0738	0.1921	0.2395	0.3430	0.4128*	0.4708*	0.5229*
	m3	0.2306	0.2598	0.2488	0.2201	0.2288	0.2403*	0.2570*
QF	n1	0.1868	0.1929	0.1989	0.2056	0.2118	0.2188	0.2270
	n2	0.2179	0.2749	0.3435*	0.3900*	0.4218*	0.4638*	0.5090*
	n3	0.2609	0.2448	0.2746	0.2681	0.2871	0.2766	0.2375*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22% confidence interval.

In Table A.1.3, we present the retrospective re-ranking retrieval performances in MAP for all the queries of the three discourse types based on the word bigrams. We put experimental results of each discourse type into separate tables (I), (II) and (III). Twelve features are used to sort the word bigram list and we have derived and introduced these features before.

Table A.1.3 Retrospective re-ranking retrieval performance based on word bigrams (Part II)

Discourse Type: Reason								
baseline		Top N of discourse type related word bigrams, N=						
=0.2241		50	100	200	500	1000	2000	5000
DFR	m1	0.1851	0.2444	0.3056	0.3929	0.426	0.441	0.4844
	m2	0.2349	0.2867	0.3240	0.3187	0.2964	0.2549*	0.2696*
	m3	0.1458	0.1921	0.2256	0.2429	0.2551	0.2593*	0.2613*
IUFR	m1	0.1174	0.1570	0.1986	0.2311	0.2480	0.2470	0.2709
	m2	0.1838	0.2283	0.3026	0.3620	0.3990	0.4293	0.4949*
	m3	0.2107	0.2222	0.2264	0.2331	0.2422	0.2483	0.2584*
WFR	m1	0.1153	0.1567	0.1873	0.2312	0.2434	0.2538	0.2709
	m2	0.1556	0.2087	0.2984	0.3538	0.3983	0.4300	0.4951*
	m3	0.2038	0.2228	0.2269	0.2330	0.2423	0.2483	0.2580*
QF	n1	0.2193	0.2219	0.2217	0.2261	0.2280	0.2296	0.2386*
	n2	0.1938	0.2310	0.2810	0.3377	0.3650	0.3992	0.4611
	n3	0.2054	0.2141	0.2352	0.2379	0.2393	0.2383*	0.2458*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.61% confidence interval.

Based the results of above tables, we can see that if we make comparison among the features based on DFR, IUFR and WFR. The latter two measures are closed and they are a little worse than the DFR. First of all, the IU size (the number of terms in an IU) is 41 in our experiments and it's not big so it's rare for a word sequence, especially

for long word sequences, to repeatedly occur in an IU. Therefore, for most of the word sequences, they occur in an IU only once. So the word sequence frequency of most of the word sequences in the IUs of a document is closed to IU frequency (the number of IUs that contain this word sequence), which is jointly caused by the size of IU and the specificity of the word sequence. Hence, the results of the features based on IUFR and WSF are closed. Second, the features based on DFR are better than IUFR as shown in above tables. Although IU frequency is more accurate than document frequency to evaluate the distribution of a word sequence due to the different sizes (lengths) of documents, IU frequency is not a good measure for the training and learning in our study. It's because we select the same number of relevant documents and irrelevant documents for balanced training.

Table A.1.3 Retrospective re-ranking retrieval performance based on word bigrams (Part III)

Discourse Type: Impact								
baseline		Top N of discourse type related word bigrams, N=						
=0.2291		50	100	200	500	1000	2000	5000
DFR	m1	0.1899	0.2334	0.2562	0.2736*	0.2705*	0.2761*	0.2740*
	m2	0.1470	0.1933	0.2766	0.3287	0.3959	0.4144*	0.4395*
	m3	0.0787	0.0964	0.0933	0.2194	0.2597	0.2877	0.2640
IUFR	m1	0.1958	0.2282	0.2459	0.2751	0.2818	0.2825	0.2898*
	m2	0.1091	0.1948	0.2623	0.3293	0.3697	0.4010*	0.4345*
	m3	0.1283	0.1838	0.2333	0.2290	0.2367	0.2417	0.2522*
WFR	m1	0.1622	0.1901	0.2135	0.2432	0.2516	0.2580	0.2690*
	m2	0.1140	0.2104	0.2947	0.3550	0.3919*	0.4287*	0.4580*
	m3	0.1901	0.2180	0.2337	0.2296	0.2293	0.2357	0.2447
QF	n1	0.2059	0.2130	0.2223	0.2259	0.2301	0.2360	0.2409
	n2	0.1507	0.2143	0.2567	0.3233*	0.3602*	0.4061*	0.4279*
	n3	0.2338	0.2405	0.2441	0.2475	0.2490	0.2464	0.2548

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.95% confidence interval.

Let us discuss the different aggregation methods for computing m_1, m_2 and m_3 features which aggregate the query level measures (e.g. query relevant document frequency) into query set level measures (e.g. query set relevant document frequency). Based on the results in the above table, m_2 is always better than the m_1 and m_3 in terms of MAP and significance testing. The aggregation methods for m_2 features (e.g. DFR m_2 , IUFR m_2) provides the best mean MAP for each query set when same number of word bigrams are used. Moreover, m_2 can make the results of re-ranking to be statistically significant with fewer word bigrams than m_1 and m_3 . The aggregation methods for m_2 features are aggregating the query level measures with a macro average-like way: the ratio of two summations without any normalization. A macro average aggregation regards each query differently and the queries with more relevant documents are more emphasized. The relevant sets of the queries within the same query set differs a lot in size, the second column of Table A.1.1 shows an example of the difference of relevant sets in our study.

As for the query frequency related features QF n_1, n_2 and n_3 , the QF n_2 is the best one in terms of the mean MAP of the re-ranking performance. Feature QF n_2 is the ratio of two total query numbers and it is similar with the aggregation method for m_2 features in that the aggregation for feature QF n_2 is a macro average-like methods. The nominator of the formula that computes feature QF n_2 is the total relevant query number that contains a word sequence and the denominator is the total irrelevant query number, see the example shown in Table A.1.1. Feature QF n_2 has good performance for sorting the word bigrams when we are looking at the number of word bigrams that can make the re-ranking result statistically significant. For discourse type

advantage/disadvantage and impact, feature QF n2 can make re-ranking result statistically significant with 200 and 500 word bigrams, which are fewer than other sorting features.

Word trigrams and word 4-grams

We have introduced the re-ranking based on word bigrams and presented the detailed experimental results by using different word sequence evaluation measures. Next we will increase the length of word sequence and word trigrams and word 4-grams are evaluated and retrospective experiments are finished with word trigrams and word 4-grams.

Let an IU contain n words, obviously we can extract $n-1$ word bigrams, $n-2$ word trigram and $n-3$ word 4-grams from it. So the total numbers of the occurrences of word bigrams, trigrams and 4-grams that can be possibly extracted are more or less the same, especially when the IU size is big. Since IU in our study contains 41 words, so from each IU we can respectively extract 39 word bigram, 38 word trigram and 37 word 4-grams. Their ratio is about 1:0.97:0.95. However, the numbers of distinct word sequences with different lengths are quite different due to their different specificity.

Table A.1.4 shows this.

Table A.1.4 Number of distinct word sequences with different lengths extracted in our study

	adv/dis	reason	impact
bigram	51952	46218	62819
trigram	82263	69032	100576
4-gram	91612	75132	112052

Generally speaking, a long word sequence is more specific than any short word

sequences that are substring of it. For example, “would pay for three” is more specific than its sub-sequence “would pay for” in sense and the former one is less likely to be matched than the latter one. Obviously, the specificity of a word sequence depends on the specificities of the words it contains as well as the sequence length. When we increase the length of the word sequences we extract from the IUs, they become more specific as a whole and we can see it from the numbers in Table A.1.4. The number of distinct word sequences increased while the total occurrence remains same, which indicates that more and more word sequence occurs for very few times. So when the length of the word sequences reaches a certain limit, most of the word sequences will occur only in one IU. Even though a long word sequence do reflects a certain discourse type, it is so hard to mine it by machine learning methods if it only occurs in one IU. Therefore, we need to find out what is the optimal range of the length of word sequence in terms of their relevance discrimination ability.

Table A.1.5 shows the distribution of word sequences with different lengths in terms of the value of DFR (Document Frequency Ratio) feature. These word sequences are extracted from the IUs of the relevant documents of eight queries of discourse type advantage/disadvantage. For every word sequence, we compute its query DFR of the eight queries by counting how many relevant and irrelevant documents containing this word sequence in the IUs are there in the relevant and the selected irrelevant set. If a query DFR of a word sequence is bigger than one, it means that more relevant documents contain this word sequence than irrelevant documents in the IUs. (It’s because the number of selected irrelevant documents is same with relevant documents, just as what we discussed before.). Query DFR is a measure to show the ability of a

word sequence to improve the retrieval.

Given a word sequence and the same number of relevant and irrelevant documents, there are three situations:

1. This word sequence occurs in the IUs of the relevant documents at least once but does not in any IU of the irrelevant documents. In this case DFR is large than one because in our formula of computing DFR the denominator is added to a small constant to avoid zero division.
2. This word sequence occurs in the IUs of the irrelevant documents but does not in any IU of the relevant documents. In this case DFR is zero since relevant document frequency is zero while irrelevant document frequency is not.
3. This word sequence occurs in the IUs of both relevant and irrelevant documents. This situation have three cases:
 - a) If DFR is bigger than one, we can say that this word sequence is more likely to occur in the IUs of relevant documents than irrelevant documents based on the data we have.
 - b) If DFR is smaller than one, we can say that this word sequence is more likely to occur in the IUs of irrelevant documents than relevant documents based on the data we have.
 - c) If DFR is equal to one, we can say that this word sequence is equally likely to occur in the IUs of irrelevant documents and relevant documents based on the data we have.

Among above situations, it is obvious that the word sequences that meet situation 1

and case (a) of situation 3 can move up the ranking positions of more relevant documents than top-ranked irrelevant documents up in the retrieved list by using this word sequence.

Then we classify all the word sequence according to the number of the values of their query DFRs that are bigger than one. There are totally eight queries so nine classes which respond to the integers from zero to eight are formed and we count the number of word sequences in each class and present them in the table. For example, among the 51592 word bigrams, there are 22 word bigrams that have six query DFRs whose value are larger than one and among 82263 word trigrams that have six query DFRs that are larger than one.

Table A.1.5 Distribution of the word sequences with different lengths in terms of the number of query DFRs that are large than one

Discourse type: Advantage/Disadvantage				
Number of queries with DFR>1	# of word bigram	# of word trigram	# of word 4-gram	# of word 5-gram
8	1	0	0	0
7	2	0	0	0
6	22	6	1	0
5	82	16	8	3
4	254	65	19	20
3	837	229	76	56
2	3404	1609	472	198
1	44642	78247	89893	92870
0	2708	2091	1143	699
Total number	51952	82263	91612	93846

From Table A.1.5, we can see that in the first seven rows (the number of query DFRs that are larger than one ranges from two to eight), with the increase of word sequence length, the number of word sequences generally decreases. That's to say,

even if we increase the length of the word sequences to six or more, we cannot expect to get more word sequences that are able to have the value of query DFR more than two. That's why the longest word sequence in following experiments is word 4-gram. The retrospective experimental results are also consistent with this decision. We will find that word bigram is the better than the word sequences with bigger lengths in improving the retrieval by our discourse type model.

Table A.1.6 Retrospective re-ranking retrieval performance based on word bigrams, trigram and 4-grams with aggregation DFR m2 sorting feature

Word Sequence Sorting Feature: DFR m2								
Disc Type		Top N of discourse type related word sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis (.2086)	bigram	.1414	.1706	.2291	.2916	.3986*	.4505*	.5098*
	trigram	.1109	.1188	.2008	.2620	.3713	.4445	.5034*
	4-gram	.0935	.1289	.1563	.2946	.3684	.4108	.5520
Reason (.2241)	bigram	.1851	.2444	.3056	.3929	.4260	.4410^	.4844^
	trigram	.1239	.2022	.2389	.3130	.3896	.4531	.5173^
	4-gram	.0760	.1573	.2109	.2753	.3423	.3959	.5479
Impact (.2291)	bigram	.1470	.1933	.2766	.3287	.3959	.4144#	.4395#
	trigram	.1435	.1950	.2443	.3634	.4013	.4473	.4849
	4-gram	.1228	.1343	.1928	.2777	.3216	.3723	.4365

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

We select DFR m2 and QF n2 as two representative word sequence sorting features in the later experiments to compare the results of using word bigrams, trigrams and 4-grams. These two features have very good performance in the past retrospective experiments based on word bigrams. Also, these two measures have

different ways to measure the distribution of a word sequence. We show the re-ranking results based on word trigrams and word 4-grams by using DFR m2 in Table A.1.6 and QF n2 in Table A.1.7.

Table A.1.7 Retrospective re-ranking retrieval performance based on word bigrams, trigrams and 4-grams with aggregation QF n2 sorting feature

Word Sequence Sorting Feature: QF n2								
Disc Type		Top N of discourse type related word sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis (.2086)	bigram	.0738	.1921	.2395	.3430	.4128*	.4708*	.5229*
	trigram	.1858	.2459	.3119	.3768*	.4139*	.5344*	.6210*
	4-gram	.0661	.0796	.1938	.2775	.3573	.4428	.5001
Reason (.2241)	bigram	.2193	.2219	.2217	.2261	.2280	.2296	.2386^
	trigram	.1182	.1764	.2386	.3178	.3842	.4304	.5182
	4-gram	.1474	.2092	.2807	.3484	.4318	.5629	.6368
Impact (.2291)	bigram	.1507	.2143	.2567	.3233 [#]	.3602 [#]	.4061 [#]	.4279 [#]
	trigram	.1669	.2216	.2738	.3401 [#]	.3819 [#]	.4384 [#]	.4975 [#]
	4-gram	.1156	.1578	.2098	.2943	.3124	.4079 [#]	.4892 [#]

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

From Table A.1.6, we find that, if sorted by the DF m2 feature, word bigram is better than word trigram and 4-gram because it can use the fewest word sequences to make the re-ranked retrieval performance statistically significantly improved with the highest confidence interval. In terms of mean MAP, the word trigram and 4-gram sometimes outperform the word bigram but the difference is not big for all three discourse types. The results of word 4-gram are not good in terms of significance test since none of the results is significantly improved. We think that it's because word 4-gram is too specific.

From Table A.1.7, we find that, if sorted by the feature QF n2, only word bigram can make the re-ranked retrieval performance statistically significantly improved with the highest confidence. Therefore, for QF n2, word bigram is also the best word sequence which is consistent with the results by using DF m2. However, we notice that for discourse type “advantage/disadvantage” and “impact”, word trigram outperforms word bigram because (1) for discourse type “advantage/disadvantage”, word trigram can make the results significant with top 500 word trigrams and while bigram requires top 1000. The mean MAP is also better than word bigram when the same number of top ranked words sequences are used. (2) For discourse type “impact”, word bigram and trigram can both make the results significant with top 500 word sequences and word trigrams are better in terms of mean MAP. This indicates that although word bigram is consistently better than other word sequences, different discourse types still have their own characters in that the best result are obtained by different word sequence distribution measures and different aggregation methods.

In summary, the retrospective re-ranking experiments based on word sequences help us to evaluate the different features of the distribution of word sequences with different lengths in the retrieved documents of a query set. These experiments also help us to investigate the performance of different aggregation methods to generate the query set-level features. Based on the results of the retrospective result, the feature DFR m2, query DFR (document frequency ratio) aggregated by the m2 method, is the best feature to provide the sorting standards for word sequences that can consistently yield good results. Also, among the query frequency features, QF n2 is also a good feature to sort word sequences, which is comparable with DF m2. Bigram is the best

length for word sequence in improving the retrieval by re-ranking and therefore to increase the length word sequence does not help.

A.2 Retrospective retrieval experiments based on POS tag sequences

In this section, we will report the results of some retrospective experiments based on POS tag sequences. These experiments are testing the measures of the POS tag sequences that we propose in section 4.3.3. In this section, we will further derive these query-level measures into query set-level features, which are obtained from a set of queries with the same discourse type. According to the value of each query set-level feature, we will sort all the extracted POS tag sequences. For each query set feature, we select a certain number of the POS tag sequences from the top of sorted list as representatives of discourse type terms. We use the same formula, just as formula A.1, to compute the re-ranked score for the retrieved documents.

For the query set features of POS tag sequences, we also propose three ways (just as the m1, m2 and m3 for word sequence) to aggregate the three measures of a POS tag sequence for a query (they are query DFR, query IUFR and query WFR) into the query set features. Let take the calculation of query set DFR as an example. We have the following three ways which correspond to m1, m2 and m3 to compute query set features for word sequences:

$$DFR_{m1}(S, ps) = \frac{1}{T} \sum_{i=1}^T DFR(q_i, ps) = \frac{1}{T} \sum_{i=1}^T \frac{RDF(q_i, ps)}{IDF(q_i, ps) + \alpha}$$

$$DFR_{m_2}(S, ps) = \frac{\sum_{i=1}^T RDF(q_i, ps)}{\sum_{i=1}^T IDF(q_i, ps)}$$

$$DFR_{m_3}(S, ps) = \frac{\sum_{i=1}^T [RDF(q_i, ps) / R(q_i)]}{\sum_{i=1}^T [IDF(q_i, ps) / I(q_i)]}$$

Above three formulas generate the feature DFR m1, DFR m2 and DFR m3 for POS tag sequence. The aggregation methods with m1, m2, m3 are also adopted to compute query set IUF and query set PSF. For the reason of brevity, we don't list the detailed formulas and they are similar with the formulas in section A.1.

We also propose three query set features based on query frequency measures. The first feature is relevant query frequency, denoted by QF n1, which is the number of queries that have a given POS tag sequence in the IUs of relevant documents. The second feature, denoted by QF n2, is the ratio of relevant query frequency to the irrelevant query frequency. In order to avoid zero division, a constant 0.1 is added for the denominator. The third feature, denoted by QF n3, is the number of queries that meet the condition that RDF of the POS tag sequence is bigger than its IFD. These three features have the similar definition with the ones in section A.1 so we don't show the formulas here.

After the POS tag sequences extracted from the IUs of the relevant documents are sorted according to one of above features, we respectively select the top 50, 100, 200, 500, 1000 2000 and 5000 POS tag sequences to build the set A (see formula 4.3.2) to re-rank the original retrieved list. After we re-rank the original retrieved list, we obtain the new MAPs and we can compare it with the original one and perform significant

tests.

In Table A.2.1, we show the number of POS tag sequences with different length that are extracted from the IUs of relevant documents of different discourse types. In order to make comparison, we also show the number of word sequences together. We can see that the number of POS tag sequence is much fewer than word sequence. The number of distinct POS tag 5-gram is close to word bigram. Our experiments are based on POS tag bigram, trigram, 4-gram and 5-gram because we find that POS tag 4-gram produces the best results and 5-gram is worse than 4-gram, which will be shown on the results in Table A.2.2. In Table A.2.2, we present the retrospective re-ranking retrieval performances in MAP for all the queries of the three discourse types based on the POS 4-grams. We put experimental results of each discourse type into separate tables (I), (II) and (III).

Table A.2.1 Number of distinct POS tag sequences extracted in our study

Discourse Type		adv/dis	reason	impact
POS Tag Sequence	bigram	1044	1042	1030
	trigram	8357	7905	8562
	4-gram	28261	24935	30603
	5-gram	54069	45579	61666
Word Sequence	bigram	51952	46218	62819
	trigram	82263	69032	100576
	4-gram	91612	75132	112052

Based the results of above tables, generally speaking, the results of the features based DFR and IUFR with the same aggregation function are very close. The best QF based feature is QF n2, which is always better than features based on DFR and IUFR. Above conclusions are consistent for all the three discourse types.

Table A.2.2 Retrospective re-ranking retrieval performance based on POS tag 4-grams (Part I)

Discourse Type: Advantage/disadvantage								
baseline =0.2086		Top N of discourse type related POS Tag 4-grams, N=						
		50	100	200	500	1000	2000	5000
DFR	m1	0.1483	0.1916	0.2273	0.2595	0.2621*	0.2701*	0.2688*
	m2	0.2509	0.3206	0.4450*	0.5386*	0.6300*	0.6545*	0.6900*
	m3	0.2660	0.3533	0.3984	0.3881	0.4179*	0.3695*	0.3234*
IUFR	m1	0.0959	0.1183	0.2034	0.2256	0.2351	0.2501*	0.2609*
	m2	0.1243	0.2456	0.4270	0.5235*	0.6249*	0.6715	0.7060*
	m3	0.2748	0.2768	0.2654	0.2693*	0.2735	0.2804	0.2903
QF	n1	0.1826	0.2045	0.2103	0.2138	0.2118	0.2149	0.2193
	n2	0.2835	0.3804*	0.4576*	0.5688*	0.6126*	0.6555*	0.6803*
	n3	0.2798	0.3143	0.3280*	0.3144*	0.3124*	0.3383*	0.2648*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22% confidence interval.

Table A.2.2 Retrospective re-ranking retrieval performance based on POS tag 4-grams (Part II)

Discourse Type: Reason								
baseline =0.2241		Top N of discourse type related POS Tag 4-grams, N=						
		50	100	200	500	1000	2000	5000
DFR	m1	0.2278	0.2502	0.2503	0.2527	0.2520*	0.2516*	0.2454*
	m2	0.2657	0.3207	0.3654	0.4428*	0.4889*	0.5091*	0.5413*
	m3	0.3001	0.3200	0.3526	0.3281	0.3090	0.2766*	0.2498*
IUFR	m1	0.1976	0.2227	0.2426	0.2554	0.2526	0.2524*	0.2502*
	m2	0.2656	0.3427	0.4024	0.4656*	0.4963*	0.5359*	0.5589*
	m3	0.2163	0.2276	0.2311	0.2323	0.2338	0.2374	0.2384
QF	n1	0.1977	0.2269	0.2241	0.2239	0.2270	0.2274	0.2317
	n2	0.2222	0.2874	0.3479	0.4368*	0.4654*	0.5006*	0.5440*
	n3	0.2256	0.2493	0.2513	0.2477	0.2448	0.2403*	0.2409*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.61% confidence interval.

**Table A.2.2 Retrospective re-ranking retrieval performance based on POS tag 4-grams
(Part III)**

Discourse Type: Impact								
baseline =0.2291	Top N of discourse type related POS Tag 4-grams, N=							
	50	100	200	500	1000	2000	5000	
DFR	m1	0.1953	0.2188	0.2326	0.2373	0.246	0.2508*	0.2549*
	m2	0.2148	0.2803	0.3437	0.3775	0.4413*	0.4702*	0.4959*
	m3	0.1101	0.1207	0.2067	0.2518	0.298*	0.2889*	0.2501
IUFR	m1	0.1958	0.2106	0.2235	0.2403	0.2510	0.2527*	0.2549*
	m2	0.1825	0.2440	0.3267	0.3898	0.4337	0.4647*	0.4865*
	m3	0.2073	0.2218	0.2296	0.2352	0.2296	0.2327	0.2379
QF	n1	0.2075	0.2103	0.2112	0.219	0.2242	0.2282	0.2345
	n2	0.1997	0.2698	0.315	0.4078*	0.4443*	0.471*	0.4955*
	n3	0.2393	0.2489	0.2468	0.2481*	0.2445*	0.2443*	0.2445*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.95% confidence interval.

As for the different aggregation methods m1, m2 and m3 which aggregate the query level measures (e.g. query relevant document frequency) into query set level measures (e.g. query set relevant document frequency). Based on the results in the above table, m2 is consistently better than the m1 and m3 in terms of MAP and significance testing. m2 aggregation methods provides the best mean MAP for each query set when same number of word bigrams are used. This conclusion is same as the conclusion we drew from the retrospective experiments based on word bigrams.

As for the query frequency related feature QF n1, n2 and n3, the QF n2 is generally the best one in terms of the mean MAP of the re-ranking performance. QF n3 feature is the second best. Feature QF n2 is generally better than feature DF m2.

Table A.2.3 Retrospective re-ranking retrieval performance based on POS tag bigram, trigram, 4-gram and 5-gram with DFR m2 sorting feature

POS Tag Sequence Sorting Feature: DFR m2								
Disc Type		Top N of discourse type related POS tag sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis (.2086)	bigram	.1339	.1895	.2709	.2585*	.2090	N/A	N/A
	trigram	.2064	.2795	.3779	.4509*	.5088*	.5435*	.3320*
	4-gram	.2509	.3206	.4450*	.5386*	.6300*	.6545*	.6900*
	5-gram	.2839	.3508	.4261*	.5254*	.5710*	.6349*	.6948*
Reason (.2241)	bigram	.2159	.2290	.2638	.2389	.2241	N/A	N/A
	trigram	.2361	.3011	.3687	.4012^	.4200^	.4460^	.2491
	4-gram	.2657	.3207	.3654	.4428^	.4889^	.5091^	.5413^
	5-gram	.2870	.3368	.3858^	.4673^	.4902^	.5393^	.5852^
Impact (.2291)	bigram	.1238	.1468	.2518	.2496	.2293	N/A	N/A
	trigram	.1938	.2379	.2968	.3522	.3740 [#]	.3964 [#]	.2838
	4-gram	.2148	.2803	.3437	.3775	.4413 [#]	.4702 [#]	.4959 [#]
	5-gram	.2076	.2667 [#]	.3326 [#]	.3991 [#]	.4558 [#]	.4772 [#]	.5117 [#]

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

In Table A.2.3 and Table A.2.4 we present the results of the respective experiments based on POS tag sequences with different lengths (from 2 to 5) with DFR m2 and QF n2 feature. Note that the numbers of POS tag bigrams for the three discourse types are all less than 2000 so we do not present the results of bigram at 2000 and 5000. We choose these two features because these two features are representative and both have the better performance than the other features. We have presented the results based on word sequences with different lengths with feature DFR m2 and QF n2. So it's also easy for us to make comparison between word sequence discourse type model and POS tag sequence discourse type model.

Let us review the re-ranking performance of POS tag sequence with different

lengths. From the results of above two tables, we find that the results of POS tag 4-gram are generally better than POS tag bigram and trigram in terms of the number of POS tag sequence that can make the re-ranking significantly improved. Compared with POS tag 4-gram, POS tag 5-gram can produce better results only for “impact” discourse type. Hence, generally, POS tag 4-gram is the best one of all.

Table A.2.4 Retrospective re-ranking retrieval performance based on POS tag bigram, trigram, 4-gram and 5-gram with QF n2 feature

POS Tag Sequence Sorting Feature: QF n2								
Disc Type		Top N of discourse type related POS tag sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis (.2086)	bigram	.1313	.1848	.2891	.2091	.2089	N/A	N/A
	trigram	.2468	.3288	.3945*	.4525*	.5184*	.5435*	.2239
	4-gram	.2835	.3804*	.4576*	.5688*	.6126*	.6555*	.6803*
	5-gram	.3034	.3791	.4656*	.5584*	.6160*	.6616*	.7186*
Reason (.2241)	bigram	.1937	.2266	.2438	.2260	.2241	N/A	N/A
	trigram	.2343	.2742	.3390	.4010^	.4173^	.4461^	.2362
	4-gram	.2222	.2874	.3479	.4368^	.4654^	.5006^	.5440^
	5-gram	.2654	.3156	.3817	.4428	.4758	.5254^	.5814^
Impact (.2291)	bigram	.1647	.2302	.2912	.3582	.3737	N/A	N/A
	trigram	.1647	.2302	.2912	.3582	.3737	.3958#	.2562#
	4-gram	.1997	.2698	.3150	.4078#	.4443#	.4710#	.4955#
	5-gram	.2275	.2762	.3448	.4164#	.4510#	.4791#	.5105#

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

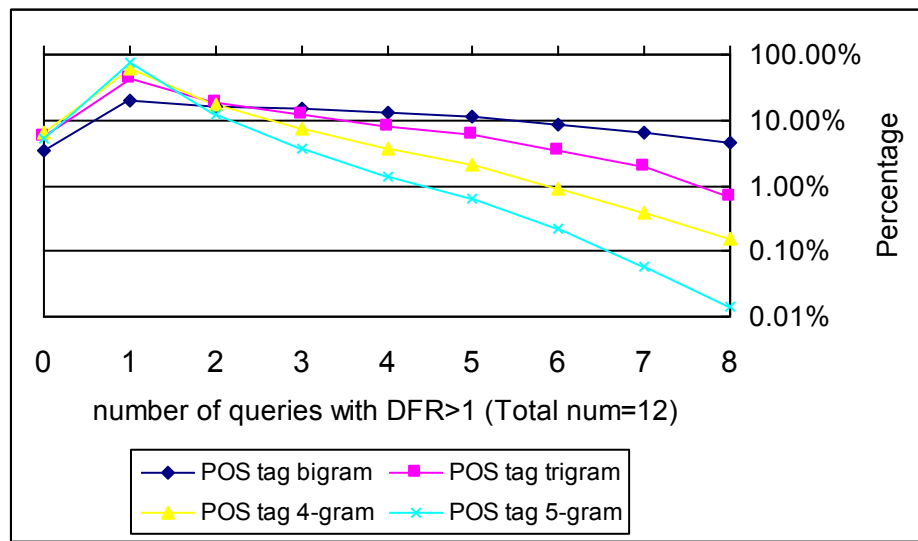
We do a statistical work on DFR feature of all the POS tag bigram, trigram, 4-gram and 5-gram for twelve queries of discourse type “impact”. We count the number of the query DRF values (twelve values in total correspond to the twelve

queries) that is larger than one and put them in the Table A.2.5. In order to make comparison among the three, we compute the percentages for POS tag sequence with four lengths and draw broken lines for the percentages in Figure A.2.1. We can see that the four lines intersect (at about 12-17%) when number of queries with DFR>1 is equal to two. After that, all the lines drop and the longer POS tag sequence is, more dramatically its line drops. It means that less POS tag sequence with good performance (in terms of query set DFR) will appear when the length of POS tag sequence increases. This is a simple analysis of the POS tag sequences with different lengths.

Table A.2.5 Distribution of the POS tag sequences with different lengths in terms of the number of query DFR values that are large than one

Number of queries with DFR>1	# of POS tag bigram	# of POS tag trigram	# of POS tag 4-gram	# of POS tag 5-gram
0	36	464	1912	3234
1	202	3766	19106	47581
2	166	1581	5114	7251
3	151	1003	2307	2193
4	130	698	1102	843
5	117	499	621	380
6	88	286	268	135
7	68	169	121	35
8	47	59	46	9
>8	25	37	6	5
Total number	1030	8562	30603	61666

Figure A.2.1 Percentage of the POS tag sequences with different lengths in terms of the number of query DFRs that are large than one for “impact” queries



In summary, the retrospective re-ranking experiments based on POS tag sequences are possible ways to evaluate the POS tag sequences with different lengths as well as the different measures of the distribution of POS tag sequences. These experiments also help us to investigate the performance of different aggregation methods for the POS tag sequences. Based on the results of the retrospective result, the DFR m2 feature, which relies on query DFR (document frequency ratio) measures with the m2 aggregation method, is the best measure to sort POS tag sequences for re-ranking. Also, among the query frequency features, QF n2 is also a good function to sort word sequences, which is comparable with DFR m2. POS tag 4-gram is the best POS tag sequence in improving the retrieval by re-ranking.

A.3 Retrospective retrieval experiments based on word-POS tag sequence

In this section, we will report the results of some retrospective experiments based on

word-POS tag sequences. These experiments are testing the measures of the word-POS tag sequences proposed in section 4.4.3. In this section, we will further derive these query-level measures into query set-level features, which are obtained from a set of queries with the same discourse type. According to the value of each query set-level feature, we will sort all the extracted word-POS tag sequences. For each query set feature, we select a certain number of the word-POS tag sequences from the top of sorted list as representatives of discourse type terms. We use the same formula, just as formula 4.2.4.1, to compute the re-ranked score for the retrieved documents.

For the query set features of word-POS tag sequences, we also propose three ways (just as the m1, m2 and m3 for word sequence and POS tag sequence) to aggregate the three measures of a word-POS tag sequence for a query into the query set features. Let take the calculation of query set DFR as an example. We have the following three ways which correspond to m1, m2 and m3 to compute query set features for word sequences:

$$DFR_{m1}(S, wps) = \frac{1}{T} \sum_{i=1}^T DFR(q_i, wps) = \frac{1}{T} \sum_{i=1}^T \frac{RDF(q_i, wps)}{IDF(q_i, wps) + \alpha}$$

$$DFR_{m2}(S, wps) = \frac{\sum_{i=1}^T RDF(q_i, wps)}{\sum_{i=1}^T IDF(q_i, wps)}$$

$$DFR_{m3}(S, wps) = \frac{\sum_{i=1}^T [RDF(q_i, wps) / R(q_i)]}{\sum_{i=1}^T [IDF(q_i, wps) / I(q_i)]}$$

Above three formulas generate the feature DFR m1, DFR m2 and DFR m3 for word-POS tag sequence. The aggregation methods with m1, m2, m3 are also adopted to compute query set IUF and query set WPSF. For the reason of brevity, we don't list

the detailed formulas and they are similar with the formulas in section 4.2.4.

We also propose three query set features based on query frequency measures. The first feature is relevant query frequency, denoted by QF_{n1} , which is the number of queries that contain the word sequence(s) that can match a given word-POS tag sequence in the IUs of relevant documents. The second feature, denoted by QF_{n2} , is the ratio of relevant query frequency to the irrelevant query frequency. In order to avoid zero division, a constant 0.1 is added for the denominator. The third feature, denoted by QF_{n3} , is the number of queries that meet the condition that RDF of the word-POS tag sequence is bigger than its IFD. These three features have the similar definition with the ones in section 4.2.4 so we don't show the formulas here.

After the word-POS tag sequences extracted from the raw and tagged text of the IUs of the relevant documents are sorted according to one of above features, we respectively select the top 50, 100, 200, 500, 1000 2000 and 5000 word-POS tag sequences to build the set A (see formula 4.4.2) to re-rank the original retrieved list. After we re-rank the original retrieved list, we obtain the new MAPs and we can compare it with the original one and perform significant tests.

In Table A.3.1, we show the number of word-POS tag sequences with different length and permutations that are extracted from the raw and tagged text of the IUs of the relevant documents of different discourse types. In order to make comparison, we also show the number of word sequences and POS tag sequences together. It's obvious that the number of word-POS tag sequences will lies between word sequence and POS tag sequence with the same length. Also, it's obvious that the number of word-POS tag sequences containing more words (e.g. "wvp" type) will be larger than the word-POS

tag sequences with the same length that contain less words (e.g. “ppw” type). In order to simplify our study, we study “pw” rather than “wp” and experiments show that these two types have very similar characters. In the same way, we study “ppw” rather than “wpp”, “wwp” rather than “pww”.

Table A.3.1 Number of distinct word-POS tag sequences extracted in our study

Discourse Type		adv/dis	reason	impact
POS Tag Sequence	bigram	1044	1042	1030
	trigram	8357	7905	8562
	4-gram	28261	24935	30603
	5-gram	54069	45579	61666
Word Sequence	bigram	51952	46218	62819
	trigram	82263	69032	100576
	4-gram	91612	75132	112052
Word-POS Tag Sequence	pw	26570	24569	31168
	pwp	42533	37302	50086
	ppw	45102	39112	52989
	wwp	67441	57699	81612
	wpw	69785	59048	83988

In Table A.3.2, we present the retrospective re-ranking retrieval performances in MAP for all the queries of the three discourse types based on the word-POS tag bigrams with “pw” types. We showed the detailed results of “pw” because it’s better than all types of the word-POS tag trigrams. We put experimental results of each discourse type into separate tables (I), (II) and (III). The comparison among word-POS tag sequences with different types will be shown later.

Based the results of above tables, we can see that if we make comparison among features based on DFR, IUFR and QF, the features based on DFR are better in terms of mean MAP and the minimum number of word-POS tag sequences that can produce

significantly improved results. The features based on IUFR are not as stable as DFR. The features based on QF are more instable, which are consistent for word sequences, POS tag sequences and word-POS tag sequences.

Among the different aggregation methods m1, m2 and m3 which aggregate the query level measures, m2 is consistently better than the m1 and m3 in terms of MAP and significance testing. This conclusion is same as the conclusion we drew from the retrospective experiments based on word sequences and POS tag sequences.

As for the query frequency related feature QF n1, n2 and n3, the QF n2 is generally the best one in terms of the mean MAP of the re-ranking performance and the minimum number of word-POS tag sequences that can produce significantly improved results, which is also consistent with the retrospective experiments based on word sequences and POS tag sequences.

Table A.3.2 Retrospective re-ranking retrieval performance based on word-POS tag bigrams with type “pw” (Part I)

Discourse Type: Advantage/disadvantage								
baseline		Top N of discourse type related “pw” type sequences, N=						
=0.2086		50	100	200	500	1000	2000	5000
DFR	m1	0.1204	0.2025	0.2429	0.2925	0.3188	0.3045*	0.3064*
	m2	0.1554	0.2050	0.3106	0.4808*	0.5538*	0.6253*	0.6919*
	m3	0.2340	0.3175	0.3415	0.3366	0.3736	0.3789*	0.3256*
IUFR	m1	0.1306	0.1650	0.2549	0.2163	0.2593	0.2848*	0.2910*
	m2	0.1508	0.2016	0.3235	0.5119*	0.5890*	0.6435*	0.7298*
	m3	0.2693	0.3045	0.2916	0.2418*	0.2290	0.2316*	0.2421*
QF	n1	0.1946	0.2000	0.2040	0.2068	0.2098	0.2119	0.2166*
	n2	0.2455	0.3121	0.4049*	0.5064*	0.5728*	0.6276*	0.7175*
	n3	0.2561	0.2808	0.3085*	0.3150*	0.3156*	0.2796*	0.3020*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22% confidence interval.

Table A.3.2 Retrospective re-ranking retrieval performance based on word-POS tag bigrams with type “pw” (Part II)

Discourse Type: Reason								
baseline		Top N of discourse type related “pw” type sequences, N=						
=0.2241		50	100	200	500	1000	2000	5000
DFR	m1	0.2471	0.2509	0.2549	0.2512	0.2568	0.2600*	0.2651*
	m2	0.2736	0.3341	0.4158	0.4980*	0.5407*	0.5800*	0.6272*
	m3	0.3206	0.3456	0.3814*	0.3637*	0.3090*	0.2446*	0.2467*
IUFR	m1	0.2188	0.2409	0.2496	0.2601	0.2579	0.2553	0.2658*
	m2	0.2438	0.3307	0.3928	0.4838	0.5291*	0.5831*	0.6320*
	m3	0.2330	0.2304	0.2304	0.2308	0.2331	0.2373	0.2429*
QF	n1	0.2208	0.2259	0.2247	0.2243	0.2253	0.2279	0.2329*
	n2	0.2196	0.2818	0.3498	0.4240	0.4989*	0.5430*	0.6190*
	n3	0.2419	0.2462	0.2373	0.2348	0.2324*	0.2326*	0.2366*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.61% confidence interval.

Table A.3.2 Retrospective re-ranking retrieval performance based on word-POS tag bigrams type “pw” (Part III)

Discourse Type: Impact								
baseline		Top N of discourse type related “pw” type sequences, N=						
=0.2291		50	100	200	500	1000	2000	5000
DFR	m1	0.2018	0.2252	0.2211	0.2416	0.2503	0.2529*	0.2572*
	m2	0.2378	0.3108	0.3618*	0.4335*	0.4614*	0.4830*	0.5093*
	m3	0.0901	0.0851	0.1351	0.2690	0.2878	0.2814*	0.2498*
IUFR	m1	0.2003	0.2254	0.2348	0.2495	0.2556	0.2593*	0.2623*
	m2	0.2228	0.2858	0.3433	0.3964*	0.4468*	0.4756*	0.5127*
	m3	0.2114	0.2460	0.2386*	0.2362	0.2353	0.2372	0.2418*
QF	n1	0.2249	0.2249	0.2263	0.2278	0.2300	0.2325	0.2368
	n2	0.2268	0.2917	0.3714*	0.4280*	0.4738*	0.4845*	0.5166*
	n3	0.2477	0.2422	0.2413*	0.2420	0.2393	0.2396	0.2414*

* indicates that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.95% confidence interval.

Table A.3.3 Retrospective re-ranking retrieval performance based on different types of word-POS tag sequences with DFR m2 sorting feature

Word-POS tag sequence Sorting Feature: DFR m2								
Disc Type		Top N of discourse type related Word-POS tag sequences, N=						
		50	100	200	500	1000	2000	5000
Adv/dis (.2086)	pw	.1554	.2050	.3106	.4808*	.5538*	.6253*	.6919*
	pwp	.1316	.2376	.3180	.4488*	.5583*	.6393*	.7083*
	ppw	.1246	.2000	.3188	.4748*	.5494*	.6068*	.6820*
	wwp	.1185	.1473	.2278	.3906	.4824*	.6163*	.6836*
	wpw	.1320	.1806	.3313	.4504	.5338	.5963*	.6741*
Reason (.2241)	pw	.2736	.3341	.4158	.4980^	.5407^	.5800^	.6272^
	pwp	.2276	.3188	.3911	.4930^	.5210^	.5797^	.6327^
	ppw	.2518	.337	.3904	.4939^	.5426^	.5950^	.6534^
	wwp	.1379	.1920	.2692	.4284	.5054	.5670^	.6996^
	wpw	.2570	.3581	.4299	.4916^	.5260^	.6011^	.6562^
Impact (.2291)	pw	.2378	.3108	.3618 [#]	.4335 [#]	.4614 [#]	.4830 [#]	.5093 [#]
	pwp	.1813	.2488	.3278	.4079 [#]	.4526 [#]	.4947 [#]	.5233 [#]
	ppw	.1998	.2680	.3364 [#]	.4183 [#]	.4638 [#]	.4941 [#]	.5324 [#]
	wwp	.1914	.2620	.3318	.4148 [#]	.4676 [#]	.4985 [#]	.5440 [#]
	wpw	.1831	.2575	.3501	.4329 [#]	.4724 [#]	.5069 [#]	.5436 [#]

*, ^ and # respectively indicate that for the queries that belong to this discourse type the difference between the results of the baseline and results of this method is statistically significant using the Wilcoxon matched-pairs signed-ranks test with 99.22%, 99.61% and 99.95% confidence interval.

In Table A.3.3 we present the results of the respective experiments based on different types of word-POS tag sequence with DFR m2 feature. One type is bigram and four types are trigrams. Compared with other features, DFR m2 has good and stable performance in the retrospective experiments on the word-POS tag sequences.

Let us review the re-ranking performance of word-POS tag sequence with different types. From the results of Table A.3.3, for the trigrams, we find that the results of “pwp” and “ppw” types are better than “wwp” and “wpw” types in terms of mean MAP and the minimum word-POS tag sequences that can produce significantly

improved results. The results of “pwp” and “ppw” are close and, if a comparison is made for the two, “ppw” is better. The only bigram “pw” is better than all the trigrams in terms of mean MAP and the minimum number of the sequences that can produce significantly improved results.

In summary, the retrospective re-ranking experiments based on word-POS tag sequences evaluate the different features on the distribution of word-POS tag sequences with different lengths and types. Based on the results of the retrospective result, the DFR m2 feature, which relies on query IUFR (IU frequency ratio) measures with the m2 aggregation method, is the best measure to sort word-POS tag sequences for re-ranking. Word POS tag bigram with type “pw” is the best word-POS tag sequence in improving the retrieval by re-ranking and second best is word-POS tag trigram with type “ppw”.

A.4 Retrieval performance after adding the discourse type terms into the queries with a new version of search engine

In order to evaluate the retrieval performance after directly adding discourse type terms into the queries, we did the experiments and showed the results in Table 4.1.3 in Section 4.1. However, we lost the records of which discourse type terms were used. We did more experiment and show the results in this section in order to justify that conclusion that directly adding discourse terms into queries cannot generally improve the retrieval performance.

The results are shown in Table A.4.1 (A, B and C) in which the columns “original”

are the baseline of using the queries shown on the left. The column “adding” in the Table A.4.1 are the results of retrieval using the queries added with discourse type term(s) which is shown at the end of each table. By comparing the results presented in column “original” and “adding”, it is obvious that adding discourse terms cannot generally improve the queries. This conclusion is consistent with the conclusion we drew from Table 4.1.3.

Table A.4.1A Retrieval performance after adding the discourse type terms into the queries

Discourse Type: advantage/disadvantage		
Topic ID and Title of Query	Original	Adding*
308 Implant Dentistry	0.125	0.077
605 Great Britain health care	0.132	0.054
608 taxing social security	0.127	0.036
624 SDI Star Wars	0.402	0.387
637 human growth hormone (HGH)	0.390	0.320
654 same-sex schools	0.001	0.006
690 college education advantage	0.004	0.001
699 term limits	0.431	0.008
Mean MAP	0.202	0.111

*Note: The discourse type terms added into the query: *advantage, disadvantage*

Table A.4.1B Retrieval performance after adding the discourse type terms into the queries

Discourse Type: reason		
Topic ID and Title of Query	Original	Adding*
333 Antibiotics Bacteria Disease	0.375	0.341
397 automobile recalls	0.495	0.441
436 railway accidents	0.176	0.193
628 U.S. invasion of Panama	0.298	0.494
636 jury duty exemptions	0.365	0.236
639 consumer on-line shopping	0.250	0.240
669 Islamic Revolution	0.048	0.049
670 U.S. elections apathy	0.206	0.188
673 Soviet withdrawal Afghanistan	0.072	0.076
Mean MAP	0.254	0.251

* Note: The discourse type term added into the query is: *reason*

Table A.4.1C Retrieval performance after adding the discourse type terms into the queries

Discourse Type: impact		
Topic ID and Title of Query	Original	Adding*
310 Radio Waves and Brain Cancer	0.055	0.040
345 Overseas Tobacco Sales	0.293	0.243
352 British Chunnel impact	0.299	0.299
391 R&D drug prices	0.153	0.148
407 poaching, wildlife preserves	0.361	0.216
448 ship losses	0.015	0.008
610 minimum wage adverse impact	0.057	0.057
641 Valdez wildlife marine life	0.488	0.472
645 software piracy	0.660	0.647
666 Thatcher resignation impact	0.008	0.008
678 joint custody impact	0.028	0.028
686 Argentina pegging dollar	0.488	0.455
Mean MAP	0.242	0.218

*Note: The discourse type term added into the query is: *impact*

REFERENCE

[Aizerman 64] M. Aizerman et al. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, v.25, pages 821–837. 1964.

[Azzopardi 06] L. Azzopardi et al. Query Intention Acquisition: A Case Study on Automatically Inferring Structured Queries. In *6th Dutch-Belgian Information Retrieval Workshop*. 2006.

[Beitzel 04] B. Steven. Hourly analysis of a very large topically categorized web query log, In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321-328, 2004.

[Berger 06] H. Berger. Exploiting partial decision trees for feature subset selection in e-mail categorization. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1105-1109, 2006.

[Blunsom 06] P. Blunsom et al. Question classification with log-linear models. In *Proceedings of the 29st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615-616, 2006.

[Boser 92] B. E. Boser, et al. A training algorithm for optimal margin classifiers. In *D. Haussler, editor, 5th Annual ACM Workshop on COLT*, pages 144-152, 1992.

[Brill 94] E. Brill. Some advances in transformation-based part of speech tagging. In *Proceedings of the twelfth national conference on Artificial intelligence*, v.1, pages 722-727, 1994.

[Broder 02] A. Broder. A taxonomy of web search. In *ACM SIGIR Forum*, v.36:2, Fall 2002

[Brooks 83] H. M. Brooks et al. Using discourse analysis for the design of information retrieval interaction mechanisms, In *Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31-47, 1983.

[Bruza 03] P. Bruza et al. A comparison of various approaches for using probabilistic dependencies in language modeling, In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 419-420, 2003.

[Buzan 96] T. Buzan. *The Mind Map Book*, Penguin Books. 1996.

[Chang 01] C. Chang et al. LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

- [Chen 01] A. Chen et al. English-Chinese cross-language IR using bilingual dictionaries. In *Proceedings of the TREC-9 Conference*, pages 15-21, 2001
- [Cooper 94] W. S. Cooper et al. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In *Proceedings of the TREC-2 Conference*, pages 57-66, 1994.
- [Cox 58] D. R. Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society, Series B (Methodological)*, v.20:2, pages 215–242, 1958.
- [Dave 03] K. Dave et al. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519-528, 2003.
- [Devijver 82] P. A. Devijver. *Pattern Recognition: A Statistical Approach*. Prentice-Hall. 1982
- [Ding 00] J. Ding et al. Computing geographical scopes of web resources. In *Proceedings of the Twenty-sixth International Conference on Very Large Databases*, pages 545-556. 2000.
- [Dombi 82] J. Dombi. A general class of fuzzy operators, the DeMorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. In *Fuzzy Sets Systems*, v. 8, page 149–163, 1982.
- [Domingos 97] P. Domingos et al. On the optimality of the simple bayesian classifier under zero-one loss. In *Machine Learning*, v.29, pages 103–130. 1997.
- [Dumais 98] S. Dumais et al. Inductive learning algorithms and representations for text categorization, In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148-155, 1998.
- [Esuli 05] A. Esuli. Determining the semantic orientation of terms through gloss classification, In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 617-624, 2005.
- [Gravano 03] L. Gravano et al. Categorizing web queries according to geographical locality, In *Proceedings of the twelfth international conference on Information and knowledge management (CIKM)*, pages 325-333, 2003.
- [Girju 02] R. Girju et al. Mining answers for causation questions. In *Proceedings of the American Association for Artificial Intelligence (AAAI)*, pages 15-25, 2002.
- [Hatzivassiloglou 97] V. Hatzivassiloglou et al. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174-181, 1997.
- [Hutchinson 05] B. Hutchinson. *The Automatic Acquisition of Knowledge about Discourse Connectives*. PhD thesis, University of Edinburgh. 2005.

[ISI 02] The ISI Question Answer Typology. Available at: http://www.isi.edu/naturallanguage/projects/webclopedia/Taxonomy/taxonomy_toplevel.html

[Jing 94] Y. Jing et al. An association thesaurus for information retrieval. In *Proceedings of RIAO 94*, pages 146-160, 1994.

[Joachims 01] T. Joachims. A statistical learning model of text classification for support vector machines, In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28-136, 2001

[Jolliffe 02] I. T. Jolliffe. Principal Component Analysis. Series: Springer Series in Statistics 2nd ed., Springer, 2002.

[Jones 02] J. Christopher et al. Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 387-388, 2002.

[Kamps 04] J. Kamps et al. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation, IV*, pages 1115-1118, 2004.

[Kang 03] I. Kang et al. Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64-71, 2003.

[Keerthi 03] S. S. Keerthi et al. Asymptotic behaviors of support vector machines with Gaussian kernel. In *Neural Computation*, v.15:7, pages 1667-1689, 2003.

[Kim 03] S. B. Kim et al. Poisson naive Bayes for text classification with feature weighting, In *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, pages 33-40, 2003.

[Knott 96] A. Knott. A data-driven methodology for motivating a set of coherence relations. PhD thesis. University of Edinburgh. 1996.

[Knott 98] A. Knott et al. The classification of coherence relations and their linguistic markers: An exploration of two languages. In *Journal of Pragmatics*, v.30, pages 135-175, 1998.

[Kwong 04] Y. K. Kong et al. Passage-based retrieval based on parameterized fuzzy operators. In *ACM SIGIR workshop on mathematical/formal methods for information retrieval*, 2004.

[Leskovec 04] J. Leskovec et al. Learning Semantic Graph Mapping for Document Summarization. In *Proceedings of PKDD04 (Conference on Principles and Practice of Knowledge Discovery in Databases)*, 2004.

- [Lewis 98] D. D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval, In *Proceedings of the 10th European Conference on Machine Learning*, pages 4-15, 1998.
- [Li 02] X. Li et al. Learning question classifiers, In *Proceedings of the 19th international conference on Computational linguistics*, pages 1-7, 2002.
- [Li 08] B. Li et al. Exploring question subjectivity prediction in community QA. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735-736, 2008.
- [Lin 01] C. Lin. On the convergence of the decomposition method for support vector machines. In *IEEE Transactions on Neural Networks*, 12(6), pages 1288–1298, 2001.
- [Lin 03] H. T. Lin et al. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, Department of Computer Science, National Taiwan University, 2003.
- [Liu 02] F. Liu et al. Personalized Web Search by Mapping User Queries to Categories. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 558-565, 2002.
- [Liu 04] H. Liu. MontyLingua: An end-to-end natural language processor with common sense. Available at: web.media.mit.edu/~hugo/montylingua.
- [Lowd 04] D. Lowd et al. Naive Bayes models for probability estimation. In *Proceedings of the 22nd international conference on Machine learning*, pages 529-536, 2005.
- [Luk 02] R. W. P. Luk et al. A Comparison of Chinese Document Indexing Strategies and Retrieval Models. In *ACM Transactions on Asian Language Information Processing*, v1, n3, pages 225-268, 2002
- [Marcu 02] D. Marcu et al. An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics Conference*, pages 368-375, 2002.
- [Martin 92] J. R. Martin. *English Text: System and Structure*. John Benjamins Pub Co. 1992.
- [Martin 07] J. R. Martin et al. *The Language of Evaluation: Appraisal in English*. Palgrave MacMillan. 2007.
- [McCallum 98] A. McCallum et al. Employing EM and Pool-Based Active Learning for Text Classification, In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 350-358, 1998
- [McCurley 01] K. McCurley. Geospatial mapping and navigation of the web. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*, pages 221-229, 2001.

- [Miranda 08] A.A. Miranda et al. New Routes from Minimal Approximation Error to Principal Components. In *Neural Processing Letters*, v.27:3, 2008.
- [Moldovan 00] D. Moldovan et al. The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the Conference of the Association for Computational Linguistics*, pages 563-570, 2000.
- [Mooney 96] R. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82-91, 1996.
- [Nigam 00] K. Nigam et al. Text Classification from Labeled and Unlabeled Documents using EM, In *Machine Learning*, v.39:2-3, pages103-134, 2000.
- [Novak 90] J. D. Novak. Concept maps and vee diagrams: Two metacognitive tools for science and mathematics education. In *Instructional Science*, v.19, pages 29-52, 1990.
- [Novak 08] J. D. Novak et al. The Theory Underlying Concept Maps and How to Construct Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008.
- [Pan 08] Y. Pan et al. Question classification with semantic tree kernel. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 837-838, 2008.
- [Pang 02] B. Pang et al. Thumbs up?: sentiment classification using machine learning techniques, In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79-86, 2002.
- [Park 02] J. Park et al. Xml Topic Maps: Creating and Using Topic Maps for the Web. Addison Wesley. 2002.
- [Pederson 01] T. Pedersen. Lexical semantic ambiguity resolution with bigram-based decision trees. In *Proceedings of the Second International Conference on Intelligent Text Processing and Computational Linguistics*, pages 157-168, 2001.
- [Pickens 06] J. Pickens. Term context models for information retrieval. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 559-566, 2006.
- [Pont 98] J. M. Ponte et al. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275-281, 1998.
- [Pu 02] H. Pu et al. Subject categorization of query terms for exploring Web users' search interests. In *Journal of the American Society for Information Science and Technology*, v.53:8, page 617-630, 2002.
- [Quinlan 93] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann

Publishers Inc., 1993.

[Raghavan 07] H. Raghavan et al. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79-86, 2007.

[Robertson 76] S. E. Robertson et al. Relevance Weighting of Search Terms. In *Journal of the American Society for Information Science*, v.27, pages 129-146, 1976.

[Robertson 86] S. E. Robertson. On relevance weight estimation and query expansion. In *Journal of Documentation*, v.42, pages 182–188, 1986.

[Robertson 94] S. E. Robertson et al. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, page 232–241, 1994.

[Robertson 97] S. E. Robertson et al. On relevance weights with little relevance information, In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 16-24, 1997.

[Santorini 90] B. Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania, 1990.

[Sowa 76] J. F. Sowa. Conceptual Graphs for a Data Base Interface. In *IBM Journal of Research and Development*, v.20:4, pages 336–357, 1976.

[Sowa 99] J. F. Sowa. Conceptual Graphs: Draft Proposed American National Standard. In *Proceedings of the 7th International Conference on Conceptual Structures: Standards and Practices*, pages 1-65, 1999.

[Stubbs 83] M. Stubbs. Discourse Analysis: The sociolinguistic analysis of natural language. University Of Chicago Press. 1983.

[Suzuki 03] J. Suzuki et al. Question classification using HDAG kernel, In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 61-68, 2003.

[Turney 02] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings 40th Annual Meeting of the Association for Computational Linguistics*, pages 417-424, 2002.

[Turney 03] P. D. Turney et al. Measuring praise and criticism: Inference of semantic orientation from association. In *ACM Transactions on Information Systems*, 21(4), pages 315-346, 2003.

[Vechtomova 03] O. Vechtomova et al. Query Expansion with Long-Span Collocates. In *Journal of Information Retrieval*, v.6, pages 251-273, 2003

- [Viterbi 67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In *IEEE Transactions on Information Theory* 13 (2): pages 260–269, 1967.
- [Voorhees 04] E. M. Voorhees. Overview of TREC 2004. In *Proceedings of the 13th Text REtrieval Conference*, 2004.
- [Voorhees 05] E. M. Voorhees. Overview of TREC 2005. In *Proceedings of the 14th Text REtrieval Conference*, 2005.
- [Wang 06] D. Y. Wang et al. An Information Retrieval Approach Based on Discourse Type. In *NLDB 2006, LNCS 3999*, pages 197-202, 2006.
- [Webber 03] B. Webber et al. Anaphora and Discourse Structure. In *Computational Linguistics*, v29:4, pages 545-587, 2003.
- [Webber 99] B. Webber et al. Discourse relations: a structural and presuppositional account using lexicalised TAG. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 41-48, 1999.
- [Whitelaw 05] C. Whitelaw et al. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625-631, 2005.
- [WordNet 06] <http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html#toc>
- [Wu 05] H. C. Wu et al. A retrospective study of probabilistic context-based retrieval, In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 663-664, 2005.
- [Wu 07] H. C. Wu et al. A retrospective study of a hybrid document-context based retrieval model, In *Information Processing and Management: an International Journal*, v.43:5, pages 1308-1331, 2007.
- [Wu 08] H. C. Wu et al. Interpreting TF-IDF term weights as making relevance decisions, In *ACM Transactions on Information Systems*, v.26:3, pages 1-37, 2008.
- [Xu 00] J. Xu et al. Improving the effectiveness of information retrieval with local context analysis, In *ACM Transactions on Information Systems (TOIS)*, v.18:1, pages 79-112, 2000.
- [Xu 08] Z. Xu. A bayesian logistic regression model for active relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 227-234, 2008.
- [Zhang 03] D. Zhang et al. Question classification using support vector machines, In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26-32, 2003.