uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Muath Alzghool

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Investigating Different Models of Cross-Language Information Retrieval from Automatic Speech Transcripts

TITRE DE LA THÈSE / TITLE OF THESIS

Diana Inkpen

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Charles Clarke (University of Waterloo)

Nathalie Japkowicz

Stan Matwin

John Oommen

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Investigating Different Models for Cross-Language Information Retrieval from Automatic Speech Transcripts

Muath Alzghool

Ph.D. Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Ph.D. of Computer Science

Under the auspices of the Ottawa-Carleton Institute for Computer Science

University of Ottawa
Ottawa, Ontario, Canada
November, 2009

Your file Votre référence
ISBN: 978-0-494-61247-7
Our file Notre référence
ISBN: 978-0-494-61247-7

Abstract

Speech information retrieval seeks to facilitate retrieving and accessing spoken content. Speech retrieval combines two techniques: automatic speech recognition (ASR) and information retrieval (IR). An ASR system is first used to transcribe digitized audio into text, and then a text retrieval system is used to retrieve speech segments, given a user request (information need). However, since ASR is an imperfect process, often there are spoken words that are not recognized correctly. This will lead to word mismatches in the retrieval. Early research considered spoken document retrieval for broadcast news as a "solved problem" [1]. However, the problem is still open for other types of speech like spontaneous conversational speech such as interviews, presentations, conferences, meetings, and lectures [2]. Unlike broadcast news (read speech), which has well-defined distinct document units that resembled written documents, spontaneous speech suffers from the lack of clear topic boundaries and poor acoustic conditions. The Cross-Language Speech Retrieval (CL-SR) track at Cross-Language Evaluation Forum (CLEF) provides a collection of oral history interviews. This offers an excellent opportunity to study different speech retrieval techniques for spontaneous speech. The availability of open source IR systems make it possible for us to investigate different Information Retrieval techniques, which proved their effectiveness in the literature for text retrieval, but they were not tested for spontaneous speech retrieval.

Moreover, we propose five novel data fusion techniques: the first one combines the results of different models with appropriate weights for each one; the second one uses a cluster-based fusion technique; the third one combines highly-varied retrieval results; the fourth fusion technique is based on a heuristic derivation of the weight for each retrieval strategy; and the last one is based on the probability theory. To deal with the word mismatch problem, we also propose two query expansion methods, one based on collocations, and the other one based on a domain-specific thesaurus.

Our system achieved the best results in the CL-SR task at CLEF for two years 2005, and 2007, and it was the second-best system in 2006.

Dedication

To my parents.

I will keep in my memory my great uncle Prof. Muhammad Raji Zughoul who passed away last year in Aug 14, 2008. He was a role model for me.

Acknowledgements

First of all, I am so thankful to Allah for giving me the power, strength, and ability to complete this work.

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

I am deeply indebted to my supervisor Dr. Diana Inkpen from the University of Ottawa, whose help, stimulating suggestions, and encouragement helped me all the time while I was researching and writing this thesis.

Prof. Charles Clarke, Prof. John Oommen, Prof. Stan Matwin and Prof. Nathalie Japkowicz deserve special thanks as my thesis committee members.

My deepest gratitude goes to my family for their unflagging love and support throughout my life; this dissertation is simply impossible without them. I am indebted to my father, Dr. Refat Alzghool, for his care and love. As a typical father in a Jordanian family, he worked hard to support the family and spared no effort to provide the best possible environment for me to grow up and attend school. He has never complained in spite of all the hardships in his life. I cannot ask for more from my mother, Fathyah Alzghool, as she is simply perfect. I have no suitable word that can fully describe her everlasting love for me. I remember many sleepless nights with her accompanying me when I was studying in high school or suffering from illness. Mom, I love you.

Also I would like to extend my appreciation to my three brothers and four sisters- Mosab, Mutasem, Muhammad, Manar, Maysa, Maram, and Malak- for their continue support and for the wonderful life when we lived together during our childhood.

I would like also to thank my grandmother, Fatimah Almasri, for her support and encouragement during my childhood; she is always asking me to be the best and to be like her oldest son, my uncle Prof. Muhammad Raji Zughoul who passed away last year; he was really a role model for me to follow when I was teenager.

Big thanks to my two sons, Almahdi and Muhammad Almutaz. They have inspired me in their own ways to finish my thesis. I want to say: "daddy has more time to play with you guys now!"

Especially, I would like to give my special thanks to my wife, Dania Qutaishat, for her endless love, encouragement, understanding, support and sacrifices which enabled me to complete this work, as a final word I would say "I love you Dandooooooooon".

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
ASR	Automatic Speech Recognition
BIR	Binary Independence Retrieval
CLEF	Cross-Language Evaluation Forum
CL-SR	Cross-Language Speech Retrieval
DFR	Divergence from Randomness
DRS	Document Retrieval System
GA	Genetic Algorithms
HMM	Hidden Markov Model
IR	Information Retrieval
LCA	Local Context Analysis
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Mean Average Precision
MALACH	Multilingual Access to Large Spoken Archives
MT	Machine Translation
NSP	Ngram Statistics Package
PRF	Pseudo Relevance Feedback
PRP	Probabilistic Ranking Principle
PR curve	Precision-recall curve
RSV	Retrieval Status Values
SDR	Spoken Document Retrieval
T	Title
TD	Title and Description
TDN	Title, Description, and Narrative
TREC	Text REtrieval Conference
TF-IDF	Term Frequency-Inverse Document Frequency

VSM	Vector Space Model
WER	Word Error Rate
ELE	Expected Likelihood Estimation

Chapter 1. Introduction

Spoken Document Retrieval (SDR) is a subfield of Information Retrieval (IR) where search techniques are applied on large collections of spoken data, in order to answer user-specified requests. With the advances of multimedia technologies, such as Automatic Speech Recognitions (ASR) systems and due to the large amount of spoken data available on the Internet (recordings from broadcast news, presentations, conferences, meetings, lectures, etc.), there is an ever-increasing interest to automatically search and browse spoken data. SDR systems facilitate fast access to spoken data, which otherwise would have to be listened to sequentially, this requiring a long time. Therefore, the amount of information that the users can access quickly is significantly increased.

Spoken document retrieval has received attention since 1997, where many research groups evaluated their systems in the speech retrieval track of the Text Retrieval Conferences TREC-6 [3], TREC-7 [4], and TREC-8 [1]. These systems were evaluated on broadcast news data. Another task was the Cross-Language Speech Retrieval (CL-SR) track at Cross-Language Evaluation Forum (CLEF), where several groups built systems for cross-language speech retrieval. Unlike in the TREC task, the systems were evaluated on conversational spontaneous speech data from 272 interviews with Holocaust survivors, witnesses and rescuers, totalling 589 hours of speech. The track was conducted for three years: 2005 [5], 2006 [6], and 2007 [7]. The CL-SR data differs from the SDR data in several ways including the lack of clear topic boundaries in conversational speech and the poor acoustic conditions.

We have participated in CL-SR track at CLEF for the three years when the track was conducted [8-10]. The focus of my thesis is on the novel techniques for information retrieval that we investigated and tested for the participation in the task and in addition to it. This chapter discusses in more detail the motivation for this work and highlights the thesis contributions.

1.1. Motivation

Spontaneous speech is an unplanned, non-rehearsed, naturally occurring, and non-experimental type of speech that forms the means of communicating information between individuals [11]. Increasing archives of digitally-recorded spontaneous speech are creating new opportunities to access the in-

formation contained in this data. However, retrieving relevant content presents significant challenges.

Currently, state-of-the-art speech recognition technologies have achieved high recognition accuracy for read texts or constrained-spoken interactions (such as broadcast news). However, accuracy is still rather poor for spontaneous speech, a speech which is not well structured and contains many disfluencies, leading to a higher error rate for automatic speech recognition systems.

We have learned from previous research done in spoken document retrieval for broadcast news that searching speech was a "solved problem", because the difference in retrieval effectiveness from error-full transcriptions generated using Automatic Speech Recognition and accurate manual transcriptions was small [1]. However, this data is read speech and it has well-defined distinct document units that resembled written documents.

Automatically-transcribed spontaneous conversational speech has special properties that make the retrieval task different than the retrieval from written text or from transcribed broadcast speech. We need to investigate more suitable techniques for this task.

The differences between transcribed spontaneous conversational speech and written text include several aspects. One comes from the nature of transcribed text which has a lot of disfluencies compared to written text. Another aspect is that the transcribed speech has a lot of miss-recognized words, due to the speech recognition errors. We believe that any successful speech information retrieval system has to deal with these issues.

The disfluencies in spontaneous speech transcriptions lead to redundant information. The four most popular disfluencies are: filler words, repetitions, repairs, and restarts. Filler words carry no information at all (i.e., "um" or "eh") or words that carry information, but not for this instance such as ("I mean" or "basically"). Repetitions are redundant pieces of information that occur when the speaker pauses for a while, considering what to say next, and then repeats the previous information. Repairs occur when the speaker says something wrong and corrects himself immediately such as (i.e., "I would like to go to France, I mean to Germany"). Restarts occur when a whole part of sentence was abandoned and the speaker starts another one. The last three disfluency types lead to redundant information which produces a different frequency distribution than the one in the equivalent written text. This is why state-of-art systems for text retrievals do not work well on speech transcripts, compared to the classical systems. For more details about the diversity of the retrieval results of these methods when applied to the spontaneous speech retrieval task, see chapters 5 and 6.

In this thesis, we have proposed different techniques to address the problems that come from the nature of spontaneous speech, and to go from written text retrieval to transcribed spontaneous speech retrieval. The first attempt is updating the stop-list (the words that contain no meaning and do not help the retrieval) in regular text retrieval systems, by including the filler words. There are two ways to address the problem of disfluencies (repetition, repairs, and restarts): the first one is to build a disfluency removal tool, and the other choice is to investigate as many weighting schemes as possible and to select the ones that work best for this task. We have focused on the second choice. We also investigated a better solution: fusing the results of different information retrieval strategies, because of the observation that the systems that worked better for written text does not necessarily work well for spontaneous text. The spontaneous speech collection that we used in our study is a about a specific domain (holocaust survivors). For this reason, we have decided to incorporate a special purpose thesaurus –which was build for this collection – in the relevance feedback methods.

Spontaneous conversational speech, where document boundaries are often not well defined, raises a number of new issues for research. To investigate these issues, the CLEF Cross-Language Speech Retrieval used data from the MALACH oral history collection to explore retrieval of spontaneous speech with significant conversational elements in the context of Cross-Language Information Retrieval [5-7]. An interesting feature of this collection is that ASR document transcriptions are accompanied by several automatically and manually derived metadata fields.

The majority of the existing research focuses on improving the recognition accuracy and performing retrieval over highly-accurate speech transcriptions for read speech, as in textual information retrieval. However, there are still some unsolved problems. In situations of low-quality acoustic conditions and low recognition accuracy, alternative approaches for improving the retrieval performance are needed. Retrieval from spontaneous speech was shown to have poor retrieval performance, due to many words that are missing from the automatic speech segments and many words that are wrongly transcribed (word mismatch problem). The numbers – especially the ones in dates in the transcribed text – are transcribed in words – cause word mismatch if the user enters the number in the query in a numeric form.

One way to address the word mismatch problem is to take advantage of multiple sources of information, such as different retrieval models. Combination of information from different sources has also been tried for textual retrieval in [5, 12-14], leading to some improvements. For the Spoken Document Retrieval (SDR) tasks at TREC [15, 16], combining the outputs of multiple systems was

shown to improve the retrieval performance. In this thesis, we have proposed several fusion techniques to combine the results of different sources of information or the results of different information retrieval strategies. For more details see chapters 6, 7, 8, and 9. The second way is to try to compensate for transcription errors by using blind relevance feedback, which also led to improvements on the TREC-SDR collection [17-26]; to achieve this goal, we have proposed several blind relevance feedback techniques in chapter 5. A third way is to combine the transcriptions with the automatically and manually-generated meta-data which is available in the CL-SR collection, but was not available in the TREC-SDR collection. The combination could be implemented during the indexing, as described in chapter 5 or after the retrieval by combining the retrieval results, as described in chapter 9. To address the number mismatch problem, a tools could be implemented to translate all the numbers in the queries or in the transcripts into corresponding words.

The difficulty in spontaneous speech retrieval is mainly caused by the style of spoken language and by the mismatch problems caused by the speech recognition system. Theses difficulties do not exist in written text. Because of this, the IR systems that work well on written text will not perform well for spontaneous speech. Therefore more investigation is required in order to address these problems.

1.2. Thesis Goals

Our research goals seek to improve retrieval effectiveness from spontaneous speech. This task is considered a difficult task due to reasons related to the properties of the data mentioned above and to other speech recognition-related properties such as: lack of clear topic boundaries; many of the important words are not articulated between participants while expressing an opinion; large number of foreign words; many languages encountered in the speech; heavy accents; emotional speech, etc. These reasons cause many challenges to the Automatic Speech Recognition System in transcribing the speech into text. Another challenge to the Information Retrieval system is to retrieve relevant transcribed segments for user-provided queries.

In the literature, there are many techniques and systems for text retrieval, which were shown to bring improvements in information retrieval from text collections or read speech collection, but the main question that we have try to answer in this thesis is: ***Which IR techniques are more suitable for spontaneous speech collections and how we could combine different techniques in order to improve the retrieval results?***

1.3. Thesis Contributions

This thesis offers four major contributions:

- Investigates different Information Retrieval techniques, which proved their effectiveness in the literature for text retrieval, but they were not tested for spontaneous speech retrieval.
- Proposes two relevance feedback methods, one based on collocations and the other one based on a thesaurus.
- Proposes five data fusion techniques, one to combine the results of as many as different retrieval strategies with reasonable weights for each strategy, the second one using a cluster-based fusion technique, the third one attempts to combine a high varied retrieval results, the fourth one attempts to derive the weight for each retrieval strategy based on heuristic fashion and the last one is a model fusion technique based on probability theories. The first three fusion methods differ in terms of the number of the retrieval strategies involved in the fusion process and in the way we select these strategies. In the first one, we select all the retrieval strategies and assigned weights to each retrieval strategy based on the recall and MAP score. This technique takes too much time, as the number of retrieval strategies is high. In the second and third fusion methods, we have proposed different ways to select the retrieval strategies, in order to save time by running fewer strategies. The class-based fusion should be used only when there is a high variation among the retrieval strategies. Finally, the probability-based model has a stronger theoretical foundation because we derived the fusion formula based on the probability theory.
- Testing of our techniques on the IR collection Multilingual Access to Large Spoken Archives Collection (MALACH), which was provided by the organizers of the CLEF CL-SR track. Our experiments reveal that the data fusion techniques are significantly better than any individual retrieval system. There are interesting properties of this collection. It contains meta-data such as manual summaries, which actually are manually-written texts. Therefore we tested our techniques for spontaneous speech transcripts, for comparison, on written as well (the manual summaries).

Our system achieved the best results in Cross-Language Speech Retrieval track at Cross-Language Evaluation Forum for two years 2005, and 2007, and it was the second best system in 2006.

We have published the following papers[8, 9, 27-32]:

1. D. Inkpen, M. Alzghool, and A. Islam, "Using Various Indexing Schemes and Multiple Translations in the CL-SR Task at CLEF 2005," *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 760–768, 2006.
2. D. Inkpen, M. Alzghool, G. Jones, and D. Oard, "Investigating Cross-Language Speech Retrieval for a Spontaneous Conversational Speech Collection," in *Proceedings of the Human Language Technology Conference / North American chapter of the Association for Computational Linguistics, HLT-NAACL 2006,* New York City, NY, 2006.
3. M. Alzghool and D. Inkpen, "Experiments for the Cross Language Speech Retrieval Task at CLEF 2006," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 778-785, 2007.
4. M. Alzghool and D. Inkpen, "Model Fusion Experiments for the Cross Language Speech Retrieval Task at CLEF 2007," *Lecture Notes In Computer Science : Advances in Multilingual and Multimodal Information Retrieval,* vol. 5152/2008, pp. 695-702, 2008.
5. M. Alzghool and D. Inkpen, "Clustering the topics using TF-IDF for model fusion," in *Proceeding of the 2nd PhD workshop on Information and knowledge management* Napa Valley, California, USA: ACM, 2008.
6. M. Alzghool and D. Inkpen, "Combining Multiple Models for Speech Information Retrieval," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)* Marrakech, Morocco, 2008.
7. M. Alzghool and D. Inkpen, "Cluster-based Model Fusion for Spontaneous Speech Retrieval," in *Proceedings of the Workshop on Searching Spontaneous Conversational Speech, SIGIR 2008* Singapore, 2008.
8. M. Alzghool and D. Inkpen, "Exploring Fusion in a Spontaneous Speech Retrieval Task," in *Proceedings of the Workshop on Searching Spontaneous Conversational Speech, ACM Multimedia 2009* Beijing, China, 2009.

1.4. Thesis Outline

This thesis is structured as follows:

- Chapter 2 introduces the preliminaries of IR research and reviews the literature in related areas such as Text Retrieval and Automatic Speech Recognition.
- Chapter 3 describe the retrieval tasks and the CL-SR collection.
- Chapter 4 reviews the literature in related areas such as CLEF's CL-SR and TREC's SDR tracks.
- Chapter 5 describes our approaches to improve the retrieval from automatic transcripts of spontaneous speech (focusing on one retrieval strategy at a time).
- Chapters 6, 7, 8, and 9 describe five novel approaches to combine the retrieval results of different retrieval strategies.
- Finally, Chapter 10 recapitulates the main contributions of the thesis and provides some directions for future work.

Chapter 2. Background Concepts and Related Work on Spoken Document Retrieval.

The field of Information Retrieval is an area of active research. Since 1950, much research was done to develop strategies and techniques for identifying the relevant documents in response to a user query. In this chapter, we present the definition of Information Retrieval (IR) and Spoken Document Retrieval (SDR), explain its key concepts and components, and outline the process by which IR operates.

2.1. Concepts of Information Retrieval

Information Retrieval is one of the first areas of natural language processing in which statistics were successfully applied. Salton [33] defined Information Retrieval (IR) as being the field concerned with the structure, analysis, organization, storage, searching, and retrieval of information. Salton's definition was very general. Most of the researchers today focus on the last aspect of Salton's definition. So, Information Retrieval has concentrated on finding relevant items from a repository of information that meet user information needs.

We can state the information retrieval problem formally as [9]:

Given a document collection D, consisting of documents d_1, d_2,...d_n, and a query Q, find all documents that are relevant to the query and return them in ranked order according to a similarity score, $SIM(Q,d_i)$, where $1<=i<=n$, and the higher values of the similarity score correspond to the most relevant documents.

Given the information retrieval problem, there are two ways to develop solutions:

1. Propose a retrieval model for measuring the similarity between documents and queries.

   Various retrieval models exist for matching documents to queries and ranking them according to similarity score [34, 35]. These include: vector space model, probabilistic model, inference net-

works, neural networks, genetic algorithms, extended Boolean model, latent semantic indexing and fuzzy set retrieval. The various approaches are discussed in Section 2.4.

2. Propose utilities techniques that could improve the retrieval which are independent of the information model used.

   The Major challenges that affect the retrieval approaches include the ambiguity of language and the concept of relevance. The ambiguity of language refers to the fact that terms used to describe a concept are not unique, or they are vague in meaning or user understanding. The same terms could describe two concepts and the same concept could be described by mutually exclusive sets of terms. To help solving this problem, various utilities techniques can be applied to information retrieval, independently of the model used. These techniques are designed to improve the effectiveness by using more accurate representations of the concepts in the document and the query. Techniques such as relevance feedback, query expansion using thesauri, stop words removal, and stemming are examples of such utilities. More details about the utilities related to our work are in Section 2.5.

Another challenge that affects retrieval is the concept of relevance. Manual review of documents by experts does not consistently result on agreement on the set of relevant documents. Even the same expert will give a different answer when asked at different times. This makes it difficult to determine which ranking is better than another. So the Text Retrieval Conference (TREC) [1] and Cross-Language Evaluation Forum (CLEF)[2] were established in 1992 and 2000 respectively to create an environment where several experts manually assess the pooled results (the top 1000 documents for each query from each participant system), in order to obtain an estimate for the correct answer set for given queries against a given document collection. The correct answer set is simply the set of document agreed to be relevant. It is neither a comprehensive set nor error free. It simply provides an independent set to help the comparison between the ranking algorithms using some evaluation measures such as Precision, Recall, MAP, etc., described in Section 3.3.

IR research has been focused on the main task of retrieving textual documents. So, for a long time, Information Retrieval was more or less synonymous with Document Retrieval or Text Re-

[1] http://trec.nist.gov
[2] http://www.clef-campaign.org

trieval. Since last decade, there is an explosion of Internet and multimedia technologies. There is a tremendous amount of data being produced and archived everyday. Among these data, large amounts of information is available in the form of spoken documents, such as recordings from broadcast news, presentations, conferences, meetings, lectures, etc. Retrieving relevant content presents significant challenges to IR systems. IR techniques for textual data are increasingly being adopted for non-textual material like spoken documents.

A Spoken Document Retrieval System (SDR) system, which is the concern of this research, is defined within the framework of Information Retrieval where requests are made of natural texts to search for information from large collections of spoken data, in order to meet user-specified requests. The output of these requests is a set of references as shown in figure 1. These references in the corpus of spoken documents will inform the end-user of possible related information of his interest. Moreover, information in spoken form can be browsed automatically. Therefore, the amount of information accessible to users can be significantly increased.

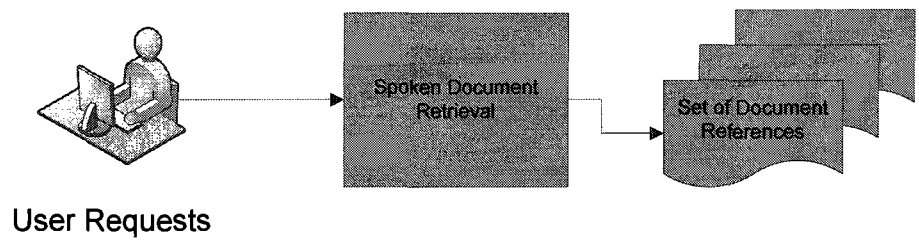

Figure 1: The Frame work of a Spoken Document Retrieval System.

Systems that are concerned with the transcription, manipulation, and retrieval of speech recordings are called Spoken Document Retrieval (SDR) systems. A typical SDR is shown in Figure 2. SDR has two main parts: an Automatic Speech Recognition (ASR) system and a Document Retrieval System (DRS). Automatic Speech Recognition is used to obtain orthographic transcriptions of the spoken documents. These transcriptions are then used to build the index for retrieval. The Document Retrieval System will retrieve and rank the documents according to the similarity to the user request. In the following sections, we will give the a brief description of the

Automatic Speech Recognizers used for automatic transcription of spoken documents as well as detailed description of the information retrieval models that are related to our studies.

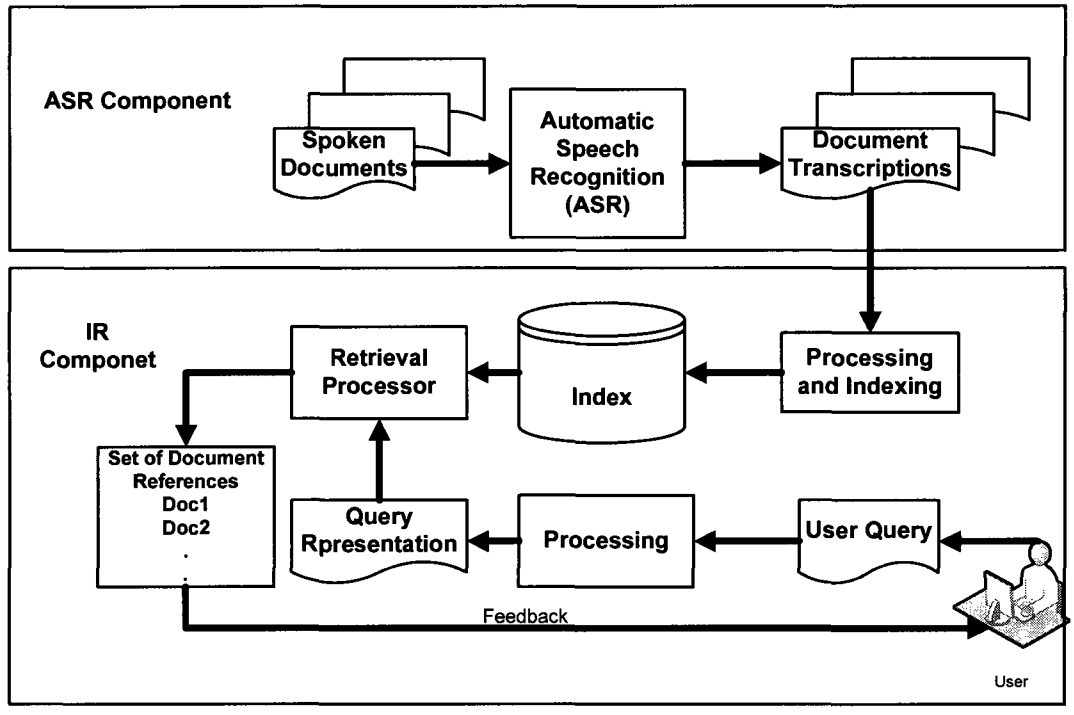

Figure 2: A Spoken Document Retrieval System.

2.2. Automatic Speech Transcription

In our research, we use the output of speech recognition (we use a speech recognizer as a black box); therefore we will talk briefly about the architecture of ASR systems.

ASR is an automated process for generating accurate written transcriptions for spoken utterances, using computer programs. ASR is useful, for example, in speech-to-text applications (dictation, meeting transcription, etc.), speech-controlled interfaces, search engines for large speech or video archives, and speech-to-speech translation.

Figure 3 illustrates the major modules of an ASR system and their relation to applications. In the feature extraction step, signal processing techniques are applied to the speech signal in order to compute the features that distinguish different phonemes from each other. Given the features extracted from the speech, acoustic modeling provides probabilities for different phonemes at different time instants. Language modeling, on the other hand, defines what kind of phoneme and word sequences are possible in the target language or application at hand, and what are their probabilities.

The acoustic models and language models are used in decoding for searching the recognition hypothesis that fits best the models and it is more likely to be the correct transcription.

The dominant technology used in ASR decoders is called the Hidden Markov Model (HMM). This technology recognizes speech by estimating the likelihood of each phoneme at contiguous, small regions (frames) of the speech signal. Each word in a vocabulary list is specified in terms of its component phonemes. A search procedure is used to determine the sequence of phonemes with the highest likelihood. This search is constrained to only look for phoneme sequences that correspond to words in the vocabulary list, and the phoneme sequence with the highest total likelihood is identified with the word that was spoken. In standard HMMs, the likelihoods are computed using a Gaussian Mixture Model; in the HMM/ANN framework, these values are computed using an artificial neural network (ANN). For more details about ASR and language modeling, see [36]. Many research groups focused their research on improving the ASR performance in order to improve the speech retrieval [1, 3, 4], for more detail about these systems, see Section 4.2.

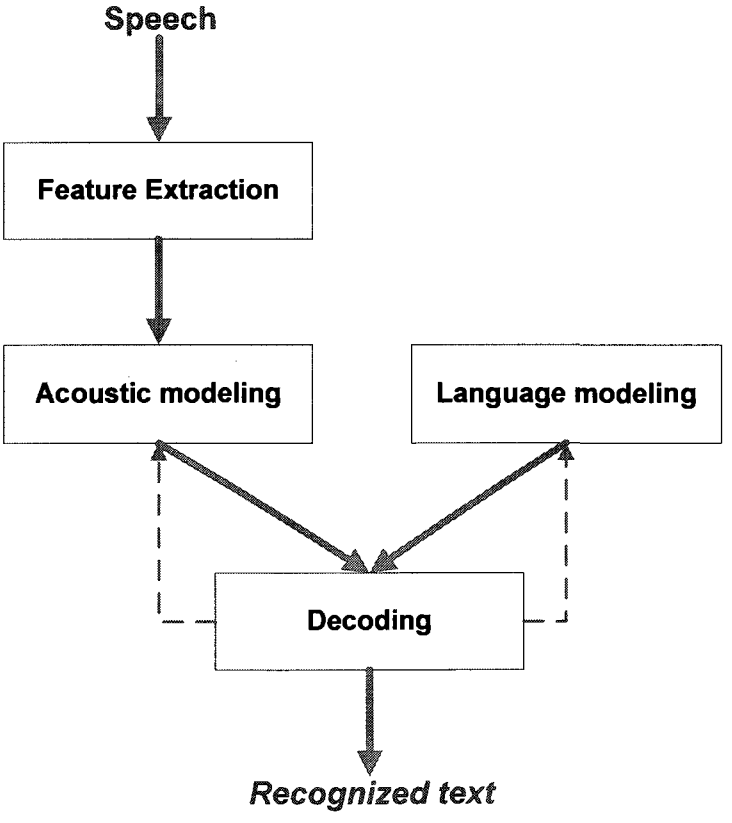

Figure 3: The general architecture of an ASR system.

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy is usually reflected by the Word Error Rate (WER), whereas speed is measured the running time. WER can then be computed as:

$$WER = \frac{S + D + I}{N} \qquad (2.1)$$

where

- S is the number of substitutions needed to align the automatic transcription with the manual reference transcription;
- D is the number of the deletions needed;
- I is the number of the insertions needed;
- N is the number of words in the reference transcription.

2.3. Document Retrieval Systems

Systems that concern the manipulation and retrieving of text documents are called Document Retrieval Systems (DRS).The three main modules of a Document Retrieval System are the input (a set of documents and one or more queries), the processing and indexing, and the output (a set of documents of interest to the user). The following is a brief description of these three sub-modules.

The Input Documents and Queries Module

Together with the document, queries are also another input for the Document Retrieval System. These inputs are originally written in natural language that the computer (until now) does not fully understand. Hence, these document and queries need to be represented in a way that can help in the process of retrieving. Many Document Retrieval Systems use the keywords list concept to achieve that. This concept is based on taking words that are considered important and ignoring other words that are considered not important (stop words) because they will not help in determining the relevancy of the documents they occur within.

The Processor and Indexing Module

The second component of a Document Retrieval System as can be seen from Figure 4 is the processing module. This module consists of two main sub-modules as shown in Figure 4 [37]. The first module is where the input document are analyzed and indexed to get their representations. These representations are then placed in a suitable file structure for further processing. The second sub-module is the one that processes the queries and matches them to document representations from the first sub-module.

Figure 5 shows the basic indexing process to derive the content representation of the document:

- Tokenization: converting a full text into a list of tokens which define the content of the text; this involves deleting markup codes, character set normalization, determining token boundaries, etc.
- Term selection: deciding which of the tokens are relevant for a content description of the document. This process usually involves at least removing stop words. A so-called stop word list usually consists of function words (conjunctions, prepositions, etc.), sometimes complemented with some high frequency words.
- Term normalization: In order to remove the redundancy which is caused by morphological variants, terms are normalized to a canonical form, a typical example is stemming.
- Term weighting: Sections 2.3- 2.6 give an elaborate overview of term weighting models.

The Output Module

The last module is the output module which normally produces a set of references to a set of documents which the system judges to be relevant to the queries.

Some of the main sub-modules that are vital for the development of a Document Retrieval System include the Stemming Module, Searching Techniques, File Structures, and Term Weight Calculation, as shown in Figure 4. The operational standard for Document Retrieval System is shown in Figure 5 [38].

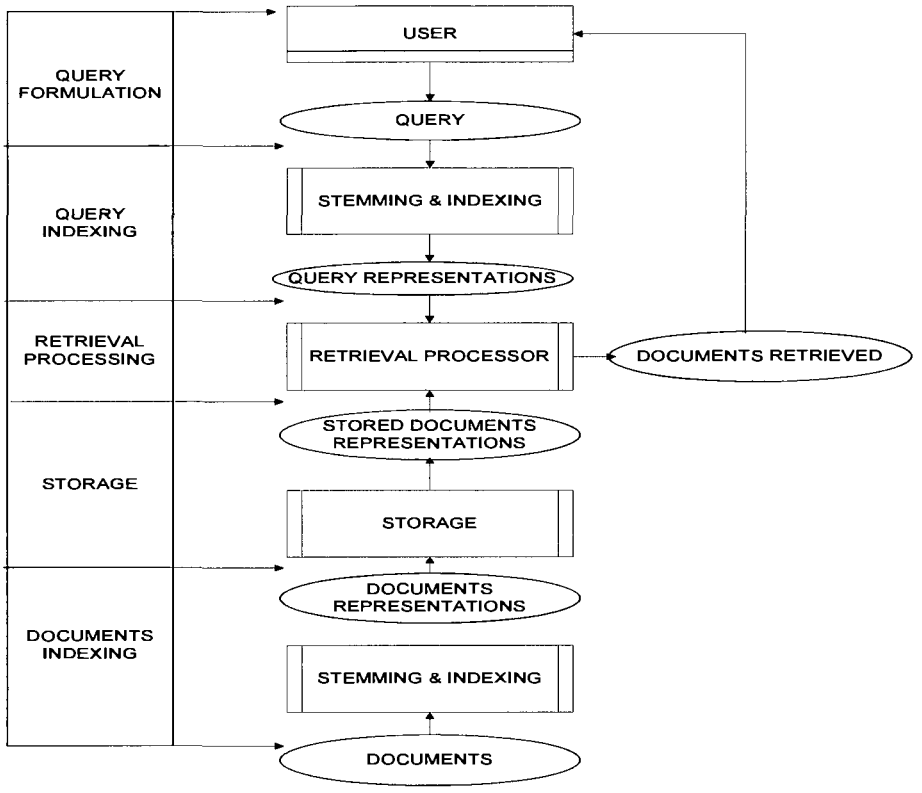

Figure 4: The general structure of a Document Retrieval System.

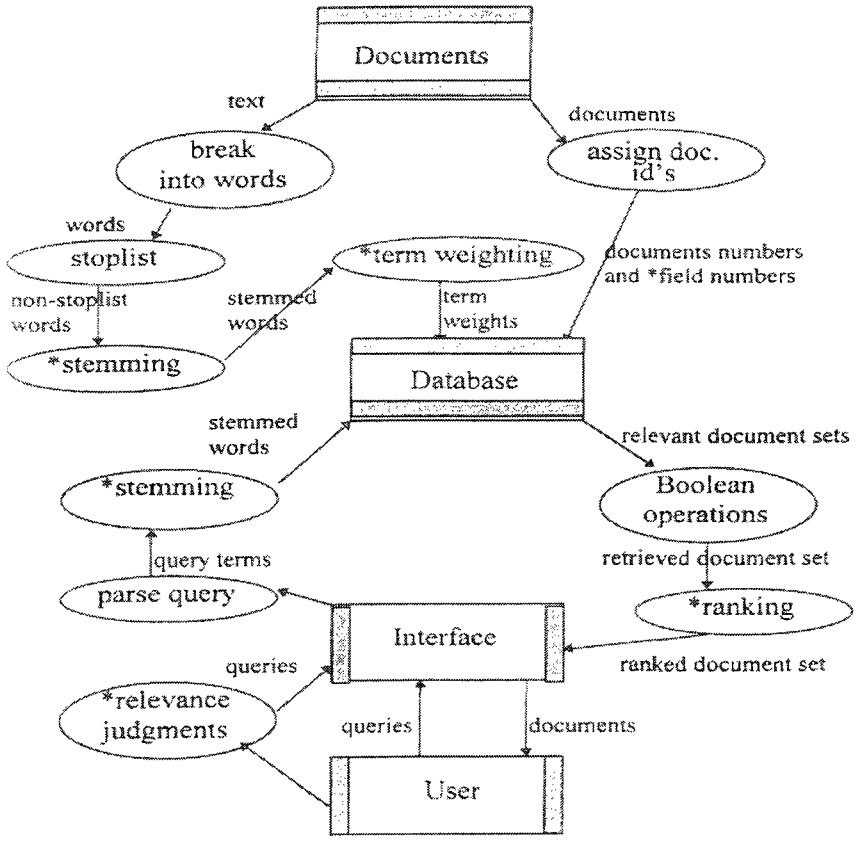

Figure 5: The operational standard of a Document Retrieval System.

2.4. Models of Information Retrieval

By information retrieval model we mean the theoretical basis of measuring the similarity between documents and queries [34]. There are several models, like Boolean model, vector space model, probabilistic model, latent semantic indexing, neural networks, genetic algorithms, and fuzzy set retrieval. There are three classic models in information retrieval system. These are the Boolean model, the vector space model, and the probabilistic model. In the Boolean model, index terms are used to represent both the document and the queries. The Boolean model is a simple

retrieval model based on the set theory and Boolean algebra [34]. The Boolean model is called set theoretic. In the vector space model, the documents and queries are represented using vectors in a t-dimensional space; consequently, this model is called algebraic. As for the probabilistic model, the framework for modeling documents and queries is based on the probability theory; thus we say that this model is probabilistic.

Our research is based on the vector space model and the probabilistic model; so we will cover them in more detail, while the other models are covered more generally. Some examples of Document Retrieval Systems that belong to the above classical models are also explained.

2.4.1 Boolean Retrieval Models

The Boolean model is a simple retrieval model based on the set theory. Due to the concept of set, the Boolean model provides a framework that is easy to grasp by a common user of a retrieval system. Furthermore, the queries are specified as Boolean expressions, which have precise semantics. The simplicity and clarity of this type of models made the Boolean models very popular as the basis for the development of commercial bibliographic retrieval systems [39]. In a Boolean retrieval system, documents are associated with a set of keywords or index terms, while queries are written in the form of a Boolean expression. The Boolean expression is a combination of the query index terms and the Boolean operators *and*, *or*, and *not*. The Boolean expression can either be entered by the user or generated by the computer from the user's natural language query [38]. The documents retrieved for a given query are those that contain index terms that match with the combination provided by the query. The Boolean model predicts or judges a document as either *relevant or non-relevant*. Hence its retrieval strategy is based on a binary decision without any possibility of ranking the documents. Consequently, the retrieval performance is much affected and a good performance is prevented.

The Boolean retrieval model is capable of giving high performance with regards to recall and precision (these terms will be discussed latter), if a good query is formulated [38]. Still, as discussed by many authors, the Boolean model has the following drawbacks [40-45]:

- The Boolean model gives counterintuitive results for certain types of queries. This limitation of the Boolean model is because of the restricted interpretations of the Boolean operators performed by the Boolean model.

- The standard Boolean model has no provision for ranking documents.
- During the indexing process for the Boolean model, it is necessary to decide whether a particular document is either relevant or not-relevant with respect to a given index term. This is because in the Boolean model there is no provision for capturing the uncertainty that is present in making indexing decision. Whereas, assigning weights to index terms adds information during the indexing process.

In spite of the above drawbacks, the Boolean model is still the basis for the development of most commercial retrieval systems due to their simplicity and to the fact that it is easy to implement efficiently [34].

2.4.2 Vector Space Model

First proposed in 1975, the vector space model (VSM) is a popular means of computing a measure of similarity between a query and a document [46]. The vector space model uses non-binary weights that are assigned to the documents and to the query index terms [43]. This will allow a partial matching retrieval instead of the relevant / non-relevant matching. The non-binary weights assigned for both the queries and documents are ultimately used to measure the degree of similarity between each of the documents stored in the system and the user query. Hence, the vector space model will also take into consideration documents which match the query terms partially.

The vector model uses t-dimensional vectors to represent both documents and queries [47]. For a document $\mathbf{d_j}$ (where j is the document number) and a query **q**, their t-dimensional vector representations are $\mathbf{d_j}$ and **q**; so the query and the document are represented by:

$$\vec{q} = \left(w_{1,q}, w_{2,q}, \cdots, w_{t,q}\right) \qquad (2.2)$$

$$\vec{d}_j = \left(w_{1,j}, w_{2,j}, \cdots, w_{t,j}\right) \qquad (2.3)$$

where $w_{i,q} \geq 0$ and $w_{i,j} \geq 0$ are the weights for the indexing term i in the query and in the document, respectively, and t is the dimension of the chosen space (the total number of indexing terms in the system).

The purpose of retrieval is to retrieve documents from the collection that match the query. In the vector space model, this is achieved by computing the similarity between the query vector

and the vectors of all the documents. Documents are then ranked in order of their measured similarities to the query. The resulting ranked list of documents is returned as the retrieval result. Users can use this ranked list of documents to locate the required information. Therefore, the definitions of the term weighting scheme and the similarity measure are two important issues in vector space models. They will be explained in greater detail as follows:

Similarity measure

The vector space model proposes to evaluate the degree of similarity of the document **dj** with regard to the query **q** as the correlation between the vectors **dj** and **q**. This correlation can be quantified, for instance, by the inner product between these two vectors [48], that is:

$$sim(d_j, q) = \sum_{i=1}^{t} w_{i,j} \times w_{i,q}$$

The model can use different similarity measures, other than the inner product, as shown in Table 1 [49]. The dot product measure is used for part of the experiments carried out in our work.

Table1 Different similarity measures used in Vector Space Models.

Similarity Measure	**Evaluation for Weighted Term Vector**
Cosine	$sim(d_j, q) = \frac{\sum_{j-1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$ (2.4)
Dice	$sim(d_j, q) = \frac{2\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2}$ (2.5)
Jaccard	$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2 - \sum_{i=1}^{t} w_{i,j} \times w_{i,q}}$ (2.6)

Term weighting scheme

Term weighting schemes are defined for the terms in query and documents to represent different emphasis for index terms in the vector space model; much work optimizing the weighting scheme was done [47, 50, 51]. Early work used manually assigned weights. A comparison of manually assigned weights and automatic weighting was done in [52, 53]. It was determined that automatic weighting performs as well as manual weighting. Index term weights can be calculated in many different ways. The most popular ways are [43]:

- Binary term weights;
- Term Frequency-Inverse Document Frequency (TF-IDF) weights.

Each term-weighting scheme from the family of TF-IDF weighting schemes is described as a combination of term frequency, collection frequency, and length normalization components [49]. The description of each component is:

▪ **Term Frequency Component**

The term frequency (TF) can be used as an indicator for term importance by assuming that frequent terms are more important. Different variation of term frequency can be considered like normalize the frequency by the maximum term frequency in the document, taking the logarithmic function to decrease the effects of large differences in term frequencies, or square the term frequency to give more importance to the term. Table 2 shows deferent ways to compute the term frequency component [49], where tf denote the term frequency of a term t in the document, $\max\_tf$ is the largest tf value in the vector, and new_tf is the new weights after applying normalizing factor.

Table 2 Term Frequency Component

Name	Shortcut	Formula
none	n	$new\_tf = tf$ (2.7)
max-norm	m	$new\_tf = \frac{tf}{\max\_tf}$ (2.8)
augmented normalized	a	$new\_tf = 0.5 + 0.5 \cdot \frac{tf}{\max\_tf}$ (2.9)
log	l	$new\_tf = \ln(tf) + 1.0$ (2.10)
square	s	$new\_tf = tf^2$ (2.11)

- **Merging of Collection Frequency Component**

The inverse document frequency (IDF) is used in the collection frequency component for discounting non-discriminative terms. Among the various terms in the documents, there exists some terms that are not useful for discriminating among the documents. For example, function words such as 'the', 'to' and 'are' have very high term frequencies, they occur in most documents and hence they have virtually no discriminating power. IDF is introduced to reflect the assumption that the discriminating power of a specific term decreases with the number of documents it occurs in. The value of IDF for a term decreases as the number of documents containing that term increases. As a result, any particular term that occurs in many documents will have a small inverse document frequency and it is assigned a small weight.

Table 3 shows different ways to compute the collection frequency component or IDF[49]. Let N denote the number of documents in the collection and df the number of documents in which term t occurs; then new_wt is defined as shown in Table 3.

Table 3 Collection Frequency Component

Name	Shortcut	Formula
none	n	$new\_wt = new\_tf$ (2.12)
inverse document frequency weight	t	$new\_wt = new\_tf \cdot \log \frac{N}{df}$ (2.13)
probabilistic	p	$new\_wt = new\_tf \cdot \log \frac{N-df}{df}$ (2.14)
squared	s	$new\_wt = new\_tf \cdot (\log \frac{N}{df})^2$ (2.15)

- **Merging of Vector Normalization**

The third component of the weighting scheme is the normalization factor, which is used to discount the effect of document lengths. It is useful to normalize the document vectors so that documents are retrieved independently of their lengths. Long documents tend to have large term frequencies, so the

size of the document will affect the similarity measure; therefore, short documents may not be recognized as relevant. Automatic information retrieval systems have to deal with documents of varying lengths in the collection. Document length normalization is used to fairly retrieve documents of all lengths [50] and it is used to remove the advantage that the long documents have in retrieval over the short documents. Table 4 shows different ways to compute the vector normalization component [49]; let *m* denote the number of entries in the vector; then the final weight *norm_wt* is defined as follows:

Table 4 Vector Normalization

Name	**Short-cut**	**Formula**
none	n	$norm\_wt = new\_wt$ (2.16)
sum	s	$norm\_wt = \frac{new\_wt}{\sum_m new\_wt}$ (2.17)
cosine	c	$norm\_wt = \frac{new\_wt}{\sqrt{\sum_m new\_wt^2}}$ (2.18)

One of the systems that we have used in our research is the SMART[3] Information Retrieval system, originally developed at Cornell University in the 1960s. SMART is based on the vector space model of information retrieval [54]. In this work we employ the notation used in SMART to describe the combined schemes: xxx . xxx. The first three characters refer to the weighting scheme used to index the document collection and the last three characters refer to the weighting scheme used to index the query fields. For example, lpc.atc means that lpc was used for documents and atc for queries. lpc would apply log term frequency weighting (l) and probabilistic collection frequency weighting (p) with cosine normalization to the document collection (c) as shown in the following formula:

$$W_{lpc} = \frac{\left(\ln(tf)+1\right)*\left(\log\frac{N-df}{df}\right)}{\sqrt{\sum_m\left[\left(\ln(tf)+1\right)*\left(\log\frac{N-df}{df}\right)\right]^2}} \qquad (2.19)$$

[3] ftp://ftp.cs.cornell.edu/pub/smart/

atc would apply augmented normalized term frequency (a), inverse document frequency weight (t) with cosine normalization (c) as shown in the following formula:

$$W_{atc} = \frac{\left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right)*\left(\log\frac{N}{df}\right)}{\sqrt{\sum_m\left[\left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right)*\left(\log\frac{N}{df}\right)\right]^2}} \qquad (2.20)$$

The main advantages of the vector model retrieval system are [34]:

- The improvement of retrieval performance due to its term-weighting scheme.
- Due to its partial matching strategy, documents that approximate the query are also retrieved.
- By using the cosine ranking measure, documents are sorted with regard to their degree of similarity with the query.

As no model is perfect, the main disadvantage of the vector model is the mutual independence of its index terms. Hence, due to locality of many term dependencies their indiscriminate application to all the documents in the collection might hurt the performance of the retrieval system.

2.4.3 Probabilistic Model

Probability theory plays a role in all studies of natural processes across all scientific disciplines. The need for a theoretical probabilistic foundation is obvious, since natural variation effects all measurements, observations and findings about different phenomena. Probability theory provides the basic techniques for statistical inference. In this section, we will talk about the basic fundamentals of probability theory with some examples, and then we will talk about different probability-based models proposed in the literature to solve IR problem.

Fundamentals of probability theory

Probability models have two essential components: sample space and probability values. The sample space (S) for an experiment is the set of all possible outcomes of the experiment, where any sub-collection of outcomes is called an event. The probabilities for each event in the sample space can be calculated or estimated.

Probabilities can be estimated in two ways [55, 56]: the first way is using a model description of the sample space and calculating the chance of occurrence of each event; the second way is data observations of the experiments.

The estimation of the probability of an event can be done by the relative frequency counts or the maximum likelihood estimation (MLE), so the probability of an event E will be computed by the following formula:

$$P(E) = \frac{number\ of\ possible\ outcomes\ of\ event\ E}{number\ of\ possible\ outcomes\ of\ sample\ space\ S} \qquad (2.21)$$

Example 1: One example of an experiment is tossing a fair coin; the sample space of the experiment is $\{head, tail\}$; we could define two events, one that represents the heads' output (E1), and another one when the output is tail (E2). Therefore p(E1) is 1/2 , and p(E2)=1/2. Note that the total probability of the sample space p(S) is 1.

Example 2: Another example of an experiment is drawing a ball from a box containing seven types of balls (4 red, 3 blue, 2 yellow, and 1 black). All balls are identical, except for their color. The sample space of the experiment is S={ red, red, red, red, blue, blue, blue, yellow, yellow, black}. Suppose we could define three events: one if the drawn ball is red E1, the second one when it is blue E2, and the last one when the ball is red or blue. The probabilities of the three events are 4/9, 3/9, 7/9, respectively.

There are three basic operations defined on the events [55, 56]: complement, union, and intersection; the complement of an event A, denoted by $\overline{A}$, occurs if and only if A does not occur; the union of two events $A \cup B$ contains all outcomes in A or B (or both); while the intersection of two events $A \cap B$ contains all outcomes which are in both A and B.

There are three basic axioms that constrain the probability space:

- The probability of an event is a non-negative real number: $P(E) \geq 0, \forall E \subseteq S$, where S is the sample space.
- The probability that some elementary event in the entire sample space will occur is 1. More specifically, there are no elementary events outside the sample space: P(S) = 1. This is often overlooked in some mistaken probability calculations; if you cannot precisely define the whole sample space, then the probability of any subset cannot be defined either.

- Any countable sequence of pair-wise disjoint events E1,E2,... satisfies $P(E1 \cup E2 \cup \cdots) = \sum_i P(E_i)$; Note: for a finite sample space, a sequence of numbers $\{p_1, p_2, p_3, \cdots, p_n\}$ is a probability distribution for a sample space $S = \{s_1, s_2, s_3, \cdots, s_n\}$, if the probability of the outcome s_k, $p(s_k) = p_k$, for each $1 \le k \le n$, all $p_k \ge 0$ and $\sum_{k=1}^{n} p_k = 1$.

There are many important rules for computing probabilities of composite events. These include conditional probability, statistical independence, multiplication and addition rules, the law of total probability, and the Bayesian rule [55, 56].

Addition Rule

The probability of a union (Addition Rule), also called the Inclusion-Exclusion principle allows us to compute probabilities of composite events represented as unions (i.e., sums) of simpler events. For events A_1 and A_2 in a probability space (S, P), the probability of the union is:

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2) \qquad (2.22)$$

Conditional Probability

The conditional probability of an event A occurring given that event B occurs is given by:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (2.23)$$

Statistical Independence

The events A and B are statistically independent if knowing whether B has occurred gives no new information about the chances of A occurring, i.e., if P(A | B) = P(A). If A and B are statistically independent, then

$$P(A \cap B) = P(A) \times P(B) \qquad (2.24)$$

Multiplication Rule

For any two events A and B (whether dependent or independent):

$$P(A \cap B) = P(A \mid B) \times P(B) = P(B \mid A) \times P(A) \qquad (2.25)$$

Law of total probability

If $\{A_1, A_2, A_3, \cdots, A_n\}$ form a partition of the sample space S (i.e., all events are mutually exclusive and $\bigcup_{i=1}^{n} A_i = S$) then the probability of any event B is

$$P(B) = P(B \mid A_1) \times P(A_1) + P(B \mid A_2) \times P(A_2) + \cdots P(B \mid A_n) \times P(A_n) = \sum_{i=1}^{n} P(B \mid A_i) \times P(A_i) \qquad (2.26)$$

Bayesian Rule

If $\{A_1, A_2, A_3, \cdots, A_n\}$ form a partition of the sample space S and A and B are any events (subsets of S), then:

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)} = \frac{P(B \mid A) \times P(A)}{\sum_{i=1}^{n} P(B \mid A_i) \times P(A_i)} \qquad (2.27)$$

Probabilistic Models for IR

The probabilistic approach to information retrieval is based on the probability theory, not on arbitrary or empirical techniques. The probability theory can be used in several ways to calculate similarity between documents and a query. Two different approaches have been proposed. The first relies on usage of patterns to predict relevance [57, 58]. The second uses each term in the query as clue to whether or not a document is relevant [59].

The original probabilistic model computes a relevance number for each document for a given query [58]. The results are ranked according to this relevance number. This model was designed for one word queries (keyword search). Each document is manually indexed with keywords. Each keyword assigned to a document is manually assigned a weight based on its value in describing the documents (the simplest weighting scheme would be binary). Given the query term, documents containing that term are reviewed and the probability that this query term would be used in search for this document is computed. This approach is based on the general likelihood of relevance of the terms. For a term t, the probability is computed as in the following formula:

$$P(t \mid \mathrm{Re}levance) / P(t \mid Non\,\mathrm{Re}levance) \qquad (2.28)$$

This equation is used as the basis for the weight of individual terms. Because of the independence assumption, one can multiply the value for each term in the document to obtain the likelihood that the document is relevant. Taking the log of the product yields:

$$sim(q, d_i) = \log\left(\prod_{i=1}^{n} \frac{P(t_i \mid \mathrm{Re}levance)}{P(t_i \mid Non\,\mathrm{Re}levance)} \right) = \sum_{i=1}^{n} \log\left(\frac{P(t_i \mid \mathrm{Re}levance)}{P(t_i \mid Non\,\mathrm{Re}levance)} \right) \qquad (2.29)$$

The assumption in Maron and Kuhan's work [58] is that the probability of this document being relevant is related to the number times this document is ever relevant to a query. So the computation is based on the probability that any request will want this document and the probability that this term will be used in a request for that document.

The classic probabilistic model was introduced in 1976 by Roberston & Sparck Jones, which later became known as the binary independence retrieval (BIR) model [59]. Most of probabilistic information retrieval models are based on the Probabilistic Ranking Principle (PRP), which says that documents should be ranked according to their probability of relevance with respect to the actual request. Our discussion is intentionally brief and focuses mainly on highlighting the key features of the model.

The probabilistic model attempts to capture the IR problem within a probabilistic framework. The fundamental idea is as follows. Given a user query, there is a set of documents which contains exactly the relevant documents and no other. This approach refers to this set of documents as the ideal answer set. Given the description of this ideal answer set, we would have no problems in retrieving its documents. Thus, we can think of the querying process as a process of specifying the properties of an ideal answer set. The problem is that we do not know exactly what these properties are. All we know is that there are index terms whose semantics should be used to characterize these properties. Since these properties are not known at query time, an effort has to be made at initially guessing what they could be. This initial guess allows generating a preliminary probabilistic description of the ideal answer set, which is used to retrieve a first set of documents. An interaction with the user is then initiated with the purpose of improving the probabilistic description of the ideal answer set. Such interaction could proceed as follows. The user takes a look at the retrieved documents and decides which ones are relevant and which ones are not (only the first top documents need to be examined). The system then uses this information to refine the description of the ideal answer set. By repeating this process many times, it is expected that such description will evolve and become closer to the real description of the ideal answer set. Thus, one should always have in mind the need to guess at the beginning the description of the ideal answer set.

The probabilistic model is based on the following fundamental assumption:

Assumption (Probabilistic Principle): Given a user query q and a document d_j in the collection, the probabilistic model tries to estimate the probability that the user will find the document dj interesting (i.e., relevant). The model assumes that this probability of relevance depends on the query and

the document representations only. Further, the model assumes that there is a subset of all documents which the user prefers as the answer set for the query q. Such ideal answer set is labeled R and should maximize the overall probability of relevance to the user. Documents in the set R are predicted to be relevant to the query. Documents not in this set are predicted to be non-relevant.

This assumption is quite troublesome because it does not state explicitly how to compute the probabilities of relevance. In fact, not even the sample space which is to be used for defining such probabilities is given.

Given a query q, the probabilistic model assigns to each document dj, as a measure of its similarity to the query, the ratio P(dj relevant-to q)/P(dj nonrelevant-to q) which computes the odds of the document dj being relevant to the query q.

For the Probabilistic model, the index term weights are all binary. A query q is a subset of the index terms. Let R be the set of documents known (or initially guessed) to be relevant. Let $\overline{R}$ be the complement of R (i.e., the set of non-relevant documents). Let $P\left(R \mid d_j\right)$ be the probability that the document d_j is relevant to the query q and $P\left(\overline{R} \mid d_j\right)$ be the probability that d_j is non-relevant to q. The similarity sim(d$_j$,q) of the document d_j to the query q is defined as the ratio

$$sim(d_j, q) = P\left(R \mid d_j\right) \Big/ P\left(\overline{R} \mid d_j\right) \qquad (2.30)$$

Using Bayes' rule and other substitutions, we find the similarity formula as:

$$sim(d, q) \sim \sum_{i=1}^{t} \left(\log \frac{P\left(k_i / R\right)}{1 - p\left(k_i / R\right)} + \log \frac{1 - P\left(k_i / \overline{R}\right)}{P\left(k_i / \overline{R}\right)} \right) \qquad (2.31)$$

which is a key expression for the ranking computation in the probabilistic model.

Since we do not know the set R at the beginning, it is necessary to devise a method for initially computing the probabilities $p\left(k_i / R\right)$ and $P\left(k_i / \overline{R}\right)$. There are many alternatives for such computation. We discuss a couple of them.

In the very beginning (i.e., immediately after the query specification), there are no retrieved documents. Thus, one has to make simplifying assumptions such as: (a) assume that $p\left(k_i / R\right)$ is constant for all index terms k_i (typically, equal to 0.5) and (b) assume that the distribution of index terms among the non-relevant documents can be approximated by the distribution of index terms among all the documents in the collection. These two assumptions yield

$$P(k_i/R) = 0.5 \qquad (2.32)$$

$$P(k_i/\overline{R}) = \frac{n}{N} \qquad (2.33)$$

where, as already defined, n is the number of documents which contain the index term k_i and N is the total number of documents in the collection. Given this initial guess, we can then retrieve documents which contain query terms and provide an initial probabilistic ranking for them. After that, this initial ranking is improved as follows.

Let R be a subset of the documents initially retrieved and ranked by the probabilistic model. Such a subset can be defined, for instance, as the top m ranked documents where m is a previously defined threshold. Further, let r be the subset of R composed of the documents in R which contain the index term k_i. For improving the probabilistic ranking, we need to improve our guesses for $P(k_i/R)$ and $P(k_i/\overline{R})$. This can be accomplished with the following assumptions: (a) we can approximate $P(k_i/R)$ by the distribution of the index term k_i among the documents retrieved so far, and (b) we can approximate $P(k_i/\overline{R})$ by considering that all the non-retrieved documents are not relevant. With these assumptions, we can write,

$$P(k_i \mid R) = \frac{r}{R} \qquad (2.34)$$

$$P(k_i \mid \overline{R}) = \frac{n-r}{N-R} \qquad (2.35)$$

This process can then be repeated recursively. By doing so, we are able to improve on our guesses for the probabilities $P(k_i/R)$ and $P(k_i/\overline{R})$ without any assistance from a human subject (contrary to the original idea). However, we can also use assistance from the user for definition of the subset R as originally conceived.

The last formulas for $P(k_i/R)$ and $P(k_i/\overline{R})$ pose problems for small values of r and R which arise in practice (such as R = 1 and r = 0). To circumvent these problems, an adjustment factor is often added in which yields

$$P(k_i \mid R) = \frac{r+0.5}{R+1} \qquad (2.36)$$

$$P(k_i \mid \overline{R}) = \frac{n - r + 0.5}{N - R + 1} \qquad (2.37)$$

An adjustment factor which is constant and equal to 0.5 is not always satisfactory. An alternative is to take the fraction n_i/N as adjustment factor.

Using the previous substitutions for $P(k_i \mid R)$ and $P(k_i \mid \overline{R})$, the similarity measure will be:

$$sim(d,q) = \sum_{t \in Q} w^{(1)} \qquad (2.38)$$

where $w^{(1)}$ is the Robertson/Sparck Jones weight [59] for the indexing term i in the query, as defined by the following formula:

$$w^{(1)} = \log\left(\frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R + r + 0.5)} \right) \qquad (2.39)$$

N is the number of documents in the collection.

n is the number of documents containing the term i.

R is the number of documents known to be relevant to a specific topic.

r is the number of relevant documents containing the term i.

The main advantage [34] of the probabilistic model is that documents are ranked in decreasing order of their probability of being relevant. The disadvantages include [34]:

- The need to guess the initial separation of documents into relevant and non relevant sets;
- The fact that the method does not take into account the frequency with which an index term occurs inside a document (i.e., all weights are binary), and
- This model only partially fits the text data because of the independent assumption of the probability. That is, the assumption that the presence of one term is independent of the presence of another term. Usually, terms found in a document are not totally independent of each other. Many terms make up phrases or are related to general concepts and are more likely to co-occur. In addition, synonyms may be less likely to co-occur. This violates the independence assumption that the probability of one term being present in a relevant document is independent of the probability of another term being present in a relevant document.

However, probabilistic information retrieval model was shown to perform well in practice. It has a very good record of accomplishment at the Text Retrieval Conference [51, 60].

Most subsequent work on probabilistic model started with Robertson and Sparck Jones' method for computing the weights and extended it to contain other important components, such as the within document frequency of the term and the relative length of the document. One of the most successful work introduced the best-match weighting function implemented in Okapi (the BM25 weighting scheme). This weighting scheme was developed at City University in 1980s and 1990s as a part of Okapi[4] information retrieval system. Okapi BM25 got the best result at the third Text Retrieval Conference TREC3 [60]. BM25 weighting is sensitive to the term frequency and the document length, as shown in the following formula. More details about how this formula can be derived are available in [51, 60, 61] :

$$sim(d,q) = \sum_{t \in Q} w^{(1)} \frac{(k1+1)tf}{K+tf} \frac{(k3+1)qtf}{k3+qtf} + k2 \cdot |Q| \cdot \frac{avdl - dl}{avdl + dl} \qquad (2.40)$$

where

K is $k1\left((1-b)+b.dl/avdl\right)$

$k1$, b, $k2$, and $k3$ are parameters which depends on the nature of the queries and possibly on the database.

tf is the frequency of the term within a specific document.

qtf is the frequency of the term within the query.

$avdl$ is the average document length.

dl is the document length.

$|Q|$ is the number of terms in the query.

Another work which extended the probabilistic model is the Divergence from Randomness (DFR) model by Amati and Van Rijsbergen [62]. This IR model ranks documents by computing the gain in retrieving a document containing a term of the query by combining two information measures Inf_1 and Inf_2 to weight the query terms according to the following formula:

$$w = Inf_1 \cdot Inf_2 = \left(-\log_2\left(Prob_1\right)\right) \cdot \left(1 - Prob_2\right) \qquad (2.41)$$

where $Prob_1$ is the probability of having by chance a tf occurrence of the term t in the document. On the other hand, $Prob_2$ is the probability of encountering a new occurrence of the term t in the

[4] http://www.soi.city.ac.uk/~andym/OKAPI-PACK/

document, given that we already found tf occurrences of the term. They have introduced five basic models to measure Inf_1:

- DLH13: generalization of the hypergeometric model in binomial case. The hypergeometric model assumes that the document is a sample, and the population is from the collection.

$$Inf1 = \log_2\left[\frac{tf \cdot avgdl}{dl} \cdot \frac{N}{F}\right] + 0.5 \cdot \log_2\left[2\pi \cdot tf\left(1 - \frac{tf}{dl}\right)\right] \qquad (2.42)$$

- DLH: another generalization of the hypergeometric model in binomial case.

$$Inf1 = \log_2\left[\frac{tf \cdot avgdl}{dl} \cdot \frac{N}{F}\right] + (dl - tf) \cdot \log_2\left[1 - \frac{tf}{dl}\right] + 0.5 \cdot \log_2\left[2\pi \cdot tf\left(1 - \frac{tf}{dl}\right)\right] \qquad (2.43)$$

- $I(F)$: Inverse term frequency.

$$Inf1 = tf \cdot \log_2\left[(N+1)/(TF+0.5)\right] \qquad (2.44)$$

- $I(n)$: Inverse document frequency where n is the document frequency

$$Inf1 = tf \cdot \log_2\left[(N+1)/(n+0.5)\right] \qquad (2.45)$$

- $I(n_e)$: Inverse expected document frequency where n_e is the document frequency expected according to a Poisson distribution.

$$Inf1 = tf \cdot \log_2\left[(N+1)/(n_e+0.5)\right] \qquad (2.46)$$

- P: Poisson approximation for the binomial distribution.

$$Inf1 = tf \cdot \log_2\left[tf/(\lambda)\right] + (\lambda - tf) \cdot \log_2 e + 0.5 \cdot \log_2\left[2\pi \cdot tf\right] \qquad (2.47)$$

- B_E: An approximation for Bose-Einstein.

$$Inf1 = -\log_2\left[N-1\right] - \log_2\left[e\right] + f\left(N+TF-1, N+TF-tf-2\right) - f\left(TF, TF-tf\right) \qquad (2.48)$$

where :

λ is the mean and variance of Poisson distribution, and it is given by $\lambda = TF / N$.

TF is the frequency of the query term t in the whole the collection.

N is the number of the documents in the whole of the collection,

n_e is given by $N \cdot \left(1 - (1 - n/N)^{TF}\right)$.

The relation f is given by Stirling formula:

$$f(n,m) = (m+0.5) \cdot \log_2\left(\frac{n}{m}\right) + (n-m) \cdot \log_2 n$$

They have also introduced two approximations for Inf_2. One based on Laplace law of succession model and the other based on the ratio of two Bernoulli processes. The two approximations provide a solution to the behavior of a statistical phenomenon called an apparent of after-effect of sampling by statisticians. It may happen that sudden repetitions of success of a rare event, such as the repeated encountering of a given term in a document, will increase the expectation of further success to almost certainty. The following formula shows the Inf_2's approximations:

$$Inf_2 \approx \begin{cases} \dfrac{1}{tf+1} & (Laplace \bmod el\ L) \\ \dfrac{TF}{n\cdot(tf+1)} & (Ratio\ B\ of\ two\ Bernoulli\ processes) \end{cases} \qquad (2.49)$$

where TF is the term frequency of the term t in the collection.

tf is the term frequency of the term t in the document

n the number of documents containing the term.

They have also proposed normalization for the tf based on the document length (dl) and the average length ($avdl$); they referred to this normalization as Normalization 2 according to the following formulae:

$$tfn = tf \cdot \log\left(1 + c \cdot \frac{avdl}{dl}\right) \qquad (2.50)$$

$$tfn_e = tf \cdot \log_e\left(1 + c \cdot \frac{avdl}{dl}\right) \qquad (2.51)$$

where dl and $avdl$ are the document length and the average length of documents in the collection, respectively. c is the free parameter of the normalization method, which can be different for different collections and it is automatically estimated [63]. tfn_e is the normalized term frequency. The logarithms are base e not 2.

For more details about the DFR models see [62]. Within the DFR framework , there are many models that were implemented in the Terrier[5] system [64]. Terrier was originally developed at Uni-

[5] http://ir.dcs.gla.ac.uk/wiki/Terrier

versity of Glasgow. The following models are implemented in Terrier to weight each term from a document that also appears in the query:

- DLH13: The DLH model is a generalization of the hypergeometric model in a binomial case. The weight was calculated according to the following formula:

$$w=\left(\log_2\left[\frac{tf\cdot avgdl}{dl}\cdot\frac{N}{F}\right]+0.5\cdot\log_2\left[2\pi\cdot tf\left(1-\frac{tf}{dl}\right)\right]\right)\cdot\left(1/(tf+0.5)\right) \quad (2.52)$$

- DLH : The DLH model is a generalization of the hypergeometric model in a binomial case. The weight was calculated according to the following formula:

$$w=\left(\log_2\left[\frac{tf\cdot avgdl}{dl}\cdot\frac{N}{F}\right]+(dl-tf)\cdot\log_2\left[1-\frac{tf}{dl}\right]+0.5\cdot\log_2\left[2\pi\cdot tf\left(1-\frac{tf}{dl}\right)\right]\right)\cdot\left(1/(tf+0.5)\right) \quad (2.53)$$

- *BB2* : Bernouli-Einstein model with Bernouli aftereffect and normalization 2.The weight was calculated according to the following formula:

$$w=\left(-\log_2[N-1]-\log_2[e]+f(N+TF-1,N+TF-tfn-2)-f(TF,TF-tfn)\right)\cdot\left(TF/n\cdot(tfn+1)\right) \quad (2.54)$$

- *BL2* : Bernouli-Einstein model with Laplace aftereffect and normalization 2.The weight was calculated according to the following formula:

$$w=\left(-\log_2[N-1]-\log_2[e]+f(N+TF-1,N+TF-tfn-2)-f(TF,TF-tfn)\right)\cdot\left(1/(tfn+1)\right) \quad (2.55)$$

- *PB2*: Poisson model with Bernouli aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w=\left(tfn\cdot\log_2\left[tfn/(\lambda)\right]+(\lambda-tfn)\cdot\log_2 e+0.5\cdot\log_2[2\pi\cdot tfn]\right)\cdot\left(TF/n\cdot(tfn+1)\right) \quad (2.56)$$

- *PL2*: Poisson model with Laplace aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w=\left(tfn\cdot\log_2\left[tfn/(\lambda)\right]+(\lambda-tfn)\cdot\log_2 e+0.5\cdot\log_2[2\pi\cdot tfn]\right)\cdot\left(1/(tfn+1)\right) \quad (2.57)$$

- $I(n)B2$: Inverse Document Frequency model with Bernouli aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w=\left(tfn\cdot\log_2\left[(N+1)/(n+0.5)\right]\right)\cdot\left(TF/n\cdot(tfn+1)\right) \quad (2.58)$$

- $I(n)L2$: Inverse Document Frequency model with Laplace aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(n+0.5)\right]\right) \cdot \left(1/(tfn+1)\right) \qquad (2.59)$$

- $I(F)B2$: Inverse Term Frequency model with Bernoulli aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(TF+0.5)\right]\right) \cdot \left(TF/n \cdot (tfn+1)\right) \qquad (2.60)$$

- $I(F)L2$: Inverse Term Frequency model with Laplace aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(TF+0.5)\right]\right) \cdot \left(1/(tfn+1)\right) \qquad (2.61)$$

- $I(n_e)B2$: Inverse Expected Document Frequency model with Bernoulli aftereffect and normalization 2. The logarithms are base 2. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(n_e+0.5)\right]\right) \cdot \left(TF/n \cdot (tfn+1)\right) \qquad (2.62)$$

- $I(n_e)L2$: Inverse Expected Document Frequency model with Laplace aftereffect and normalization 2. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(n_e+0.5)\right]\right) \cdot \left(1/(tfn+1)\right) \qquad (2.63)$$

- $I(n_e)C2$: Inverse Expected Document Frequency model with Bernoulli aftereffect and normalization 2. The logarithms are base e. The weight was calculated according to the following formula:

$$w = \left(tfn \cdot \log_2\left[(N+1)/(n_e+0.5)\right]\right) \cdot \left(TF/n \cdot (tfn_e+1)\right) \qquad (2.64)$$

Another probabilistic model, not based on the concept of relevancy, was proposed by Ponte and Croft [65, 66]. This model was based on Language Model (LM), where the probability that the query was generated from the document model, by assigning a score for each document. Lemur6 is an information retrieval system based on language models.

[6] http://www.lemurproject.org/lemur/overview.php

2.4.4 Other Models

Many other strategies for retrieval exist. In this section, we briefly discuss some additional retrieval strategies, for the sake of completeness. None of the techniques discussed bellow are used in our experiments.

Fuzzy Set Retrieval

A fuzzy set is a set where each element has a degree of membership (a weight), rather than a binary membership, i.e., member or not member. For information retrieval, a set may be defined for each document with the terms as the members [45, 67, 68]. The weight of membership may be defined using the TF-IDF measure. This is very useful in terms of indicating which terms are more important than others in distinguishing between documents. To compute similarity, the union operator is used for an OR and the intersection for an AND. The set functions are modified for the fuzzy set to be defined as the maximum weight for a union operator, and the minimum weight for the intersection operator. More formally, $a \cup b = \max\left(weight(a), weight(b)\right) \in C$ where $weight(a)$ is the weight of membership in a and $weight(b)$ is the weight of membership in b. Likewise, $a \cap b = \min\left(weight(a), weight(b)\right) \in C$.

Inference Networks

Inference networks for information retrieval are an extension of the probabilistic model. However, this strategy incorporates inferential relationships among documents, terms, and queries [69]. Inference Networks use evidential reasoning to estimate the probability that a document will be relevant to query. Beginning with one document and one query, the links from the terms in the documents instantiate the term nodes in the term layer. These nodes are given a weight such as their TF-IDF score. These, in turn, activate the links from those terms to the query, and the query node is activated. The query node then computes the belief in the query, qiven this document, and a computation is used as a similarity coefficient for ranking the document.

Latent Semantic Indexing

While most approaches to information retrieval match directly on query terms, latent semantic indexing does not require the actual presence of the query term in a document of relevance. Given that a specific concept may be described by various terms in language, this represents a strong advantage. The Latent Semantic Indexing introduces a matrix of terms to documents and uses a singular value decomposition to represent the "latent semantics" of the documents, regardless of the terms [70, 71]. The Latent Semantic Indexing was shown to perform slightly better the vector space model [72]. However, this approach is extremely computationally expensive and does not scale well to large collections.

Genetic Algorithms

Genetic Algorithms (GA) are computational algorithms based on the genetic model of evolution. Beginning with an initial population, members are evaluated, some are selected for modifications, such as mutations and crossovers, and reproduction occurs, creating the next generation of the population; then the process of reproduction is repeated several times. This approach has been used to implement information retrieval in a variety of ways, including determining the best representation of the document [73], the best weights for query terms [74], and the best formulation of Boolean queries [75]. This work was mostly done on small collections, with few queries.

Neural Networks

Neural Networks were used in information retrieval in several ways: to implement the vector space model [76], to implement the probabilistic model [77], to implement relevance feedback, and to implement learning models to improve the above models [78]. Three types of nodes are used to set up a neural network for information retrieval: QUERY, TERM, and DOCUMENT. The links between the nodes are defined as query-term links and document-term links. A query-term link indicates that the term occurs in the query. The weight of the link is defined as the TF-IDF score for the vector space model, and the Sparck-Jones weight for the probabilistic model. Similarly, a document-term link indicates that the term exists in a document, with an appropriate weight. A feed-forward network is implemented, which activates a given node when the output of the node exceeds a given threshold.

2.5. Techniques to Enhance the Basic Information Retrieval System

In this section, we will discuss techniques that are frequently applied to improve IR performance, but are often viewed as external to the retrieval models discussed in the previous section. We will present an overview to the techniques used in our experiments. This section starts with a discussion of relevance feedback, which is a technique for enhancing a query on the basis of relevancy information. The feedback information can either be used to re-weight query terms or to expand the query with terms from relevant documents. The latter technique can also be used to improve noisy document representations, e.g., for speech transcripts. Stop word removal and stemming are used to improve recall and document representation. Stemming conflate related words to index word. A stop word list contains a list of words that fail to discriminate between documents. The last technique which is used to improve the retrieval performance is data fusion, which combines the search results from multiple search techniques.

2.5.1 Query Expansion and Relevance Feedback

If a relevant document does not contain the terms that are in the query, then that document will not be retrieved. The aim of query expansion is to reduce this query/document mismatch by expanding the query using words or phrases with a similar meaning, or some other statistical relation to the set of relevant documents. There are many techniques for query expansion; they can be classified into manual relevance feedback; pseudo relevance feedback (PRF), thesauri techniques, and local context analysis. In this section, we briefly explain each approach. Some of these techniques are used in our experiments.

Relevance Feedback

Relevance feedback is a technique used to improve information retrieval effectiveness by augmenting a query with data from documents known or assumed to be relevant. The basic approach is to use multiple passes to find the best list of relevant documents; then the system will generate a new list of retrieved documents. If the user is involved into the selection of the relevant documents, this technique is called manual relevance feedback. The first one who proposed this technique was Rocchio [79, 80]. It was originally implemented with manual feedback: after the user judged the relevant documents, Rocchio used the vector space model and modified the query vector to add to it the sum of the vectors for all relevant documents, normalized by dividing by the number of documents. He

used a similar process for those documents judged to be non relevant; the sum of the vectors of the non-relevant documents was subtracted from the resulting query vector. Weights were added to give the original query terms the highest weights, along with the positive and negative weights, to adjust the impact of the feedback. This process could be repeated with the results of the query as many times as desired. The Rocchio's formula for relevance feedback is shown bellow:

$$\vec{q}_m = \alpha \vec{q}_o + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

where $\vec{q}_m$ is the modified query, $\vec{q}_o$ is the original query vector, D_r and D_{nr} are the set of known relevant and non-relevant documents, respectively, and α, β,and γ are weights attached to the term. Rocchio focused on the impact of the relevant and non-relevant document weight multipliers β and γ.

Subsequent work showed that the best results were achieved when only one non-relevant document was subtracted from the query vector [81]. Nonetheless, many systems eliminate the negative component altogether and only add terms with positive weights. For example, Salton and Buckley showed improvements using relevance feedback in the SMART system, by setting the weight of non-relevant documents to zero [81], but this improvement was less than when the setting was $\beta = 0.75$ and $\gamma = 0.25$.

Another improvement was achieved by using the lowest-ranked 500 documents for automated negative feedback [50].

Relevance feedback was implemented using the probabilistic model as well [81-87]. This approach uses a probability of relevance given the term is present in the document and the probability of non-relevance when the term is present.

The process of relevance feedback was extended to eliminate the user from the loop by automatically assuming the top x documents returned as relevant. This is called pseudo-feedback. Work has shown that best results are achieved when not all the terms from the top documents are used [88]. For instance, the top terms (based on normalized IDF or another measure for term quality) from the relevant documents are added to the original query terms, and the query is rerun. Buckley added the most frequently-occurring 50 single terms and 10 phrases from the top 20 documents; the terms in the query were re-weighted using the Rocchio formula with $\alpha = 1, \beta = 1$, and $\gamma = 0$ [88]. Many approaches to weighting the feedback terms were introduced. Often, feedback term weights are dis-

counted by a factor such as 0.5. The INQUERY system has a weighting scheme based on the rank of the feedback term in the list of possible feedback terms: the lower the rank, the higher the discount [19]. Harman experimented with term re-weighing by query expansion, feedback term selection techniques or sort orders, and the effectiveness of performing multiple iterations of relevance feedback [87]. Harman determined that using only selected terms for relevance feedback produced better results than using all the terms from the relevant documents. She also determined that using a feedback term sort order that incorporated information about the frequency of the term in the document, as well as the term's overall collection frequency, produced improved results. Landquist also showed that the using the best 10-20 terms and phrases from the top 5-20 documents with a scaling factor of 0.5 for the terms added to the query performed best [25].

Local and global context analysis is a specialized form of relevance feedback where information about term occurrence across the top retrieved documents is used to select terms and to adjust the weights of the terms. Examples of local information that might be used in feedback include: the number of top document that contain the term; the number of passages from the top documents where the candidate term co-occurs with a query term; and local-IDF, where the inverse document frequency is recalculated using the retrieved set instead of the entire collection. [84] used local context information to modify the weights of the query terms without adding additional query terms. [89] implemented local context analysis on TREC3, TREC4, and Tipster 2 and 3 collections.

Another approach which used the local and global context analysis was proposed by Amati in 2003 [90]. Amati's relevance feedback model is based on the Bose-Einstein statistics. He weights each term in the top-ranked documents according to Table 5.

Table 5 Terrier's query relevance feedback weight for each term in the top ranked documents.

Model	**Formula**
KL	$w_{KL}(t) = P_x \cdot \log_2 \frac{P_x}{P_c}$ (2.65)
Bo1	$w_{Bo1}(t) = tf_x \cdot \log_2 \frac{1+P_n}{P_n} + \log_2(1+P_n)$ (2.66)
Bo2	$w(t) = tf_x \cdot \log_2 \frac{1+P_f}{P_f} + \log_2(1+P_f)$ (2.67)

The notation in Table 5 is explained bellow:

tf_x is the frequency of the query term in the top-ranked list.

l_x is the sum of the length of exp_doc top-ranked list.

$token_c$ is the total number of tokens in the whole collection.

P_n is given by TF/N where TF is the term frequency of the query term in the whole collection and N is the number of documents in the whole collection.

P_f is given by $\frac{tf_x \cdot l_x}{token_c}$

P_x is given by $\frac{tf_x}{l_x}$

and P_c is given by $\frac{TF}{token_c}$

Amati employed two alternative methods to determine qtw, the query term weight of a re-weighted term. The first method uses the Rocchio's beta formula [1, 12]:

$$qtw = \frac{qtf}{qtf_{max}} + \beta \cdot w \frac{w(t)}{w_{max}(t)} \qquad (2.68)$$

where $w(t)$ is the weight of term t and $w_{max}(t)$ is the maximum $w(t)$ of the expanded query terms. β is a parameter.

The other one is a parameter-free method, where the qtw of a re-weighted term is given by:

$$qtw = TF_{max} \cdot \log_2 \frac{1 + P_{n,max}}{P_{n,max}} + \log_2 \left(1 + P_{n,max}\right) \qquad (2.69)$$

where $P_{n,max}$ is given by TF_{max}/N and TF_{max} is the TF of the term with the maximum $w(t)$ in the top-ranked documents.

Thesaurus Query Expansion

A thesaurus is generated automatically from the text collection, or by manual methods. The thesaurus can be used to expand the queries, in order to improve the coverage of the retrieved results. Semantic Networks or concept hierarchies are structures in which individual concepts are linked to

other related concepts. WordNet is the most widely-known general semantic network. There are many term relations in a thesaurus, such as more specific, less specific, synonym, etc. Term selection for query expansion is based on the desired relation [91]. For instance, a query for the term "dog" could be expanded using the more specific relation to add the terms "hound", "pit-bull", "terrier", etc. There is little evidence that a general thesaurus can be consistently effective in query expansion [92]. More success has been achieved with global context analysis techniques that generate a domain-specific thesaurus from text collection. Sparck Jones used co-occurrence of terms to create clusters of terms used for expansion [93]. Other works use the set of n words surrounding the search term, for a more specific selection of the expansion terms [94-96].

2.5.2 Stop Words and Stemming

Both stop words and stemming are common techniques used to improve the document representation in information retrieval. Stemming conflates related words into one word (i.e., "compute" and "computes"); this improves term matching with the query words. Removing stop words is helpful in improving the speed of the retrieval, reducing the size of the index, and it was proven to increase the effectiveness of the retrieval.

Stop words refer to those words used in documents that convey little or no meaning and consists of frequent words or words that fail to discriminate between documents. According to the Brown corpus, the most commonly used words in written English are: "the", "of", "and", "to", "a", "in", "that", "is", "was", and "he". These ten words make up 25% of written text. The top 100 words make 40% of text [97]. These words are so common in all documents that the do not help discriminate relevant documents from non-relevant documents. In addition, they increase the storage requirements of an information retrieval system by adding index terms for each document. Such words are generally listed in a stop list indicating that they should not be indexed when they appear in the source text and should be skipped. Prior work in developing and testing stop lists indicate that stop lists do not degrade the effectiveness of information retrieval, but in some case they improve it [98, 99]. A document-document similarity calculation identifies words that are not differentiators for query results. This approach removed 75% of the terms in the collection but it is so computationally-intensive that it is impractical for large collections. The most commonly-used stop list is the manually-generated list used in the SMART system[7]. Lo proposed different techniques to generate a list of stop words

[7] ftp://ftp.cs.cornell.edu/pub/smart/

automatically by determining how informative a term is by using the Kullback-Leibler divergence measure[100]. In our experiments, we have used an updated version of the SMART system, by extending the list of stop words with filler words which carry no information (i.e., "um" or "eh") or words that carry information but not relevant to the respective text (such as "I mean" or "basically"). These filler words do usually not appear in regular written text. So, we are changing the usual stop-list to become appropriate for spontaneous speech retrieval.

Stemming is a technique that conflates word variants into one entry in the information retrieval index. For example, ending such as -s, -es, -ing, -ly are removed from the terms. In this way, the terms "train" and "training", for example, get indexed as the same term. This is very helpful in broadening a search beyond the specific term variants found in the query. However, it is dangerous in that ambiguous terms and unrelated terms get conflated. If "train" referred to a connected series of vehicles and "training" to acquisition of knowledge, they are erroneously joined through stemming. Several techniques deal with word sense disambiguation in information retrieval [101-103]. Several stemmers algorithms exist [104, 105], The first published stemmer was written by Lovins in 1968 [106]. The most popular stemmer is the Porter Stemmer [107].

2.5.3 Fusion of Various Retrieval Strategies

IR research on data fusion studies the combination of search results from multiple search techniques. There are many situations where data fusion will be very helpful:

- It is practical when many IR systems are available, and none of them are superior to the other systems in all cases. We aim to provide better results than the best individual system.
- It is very suitable when each system has it is own database, where each system may not have the entire data set.

In these situations, many techniques for data fusion could be implemented, since many IR systems are available for research like: SMART[8], Terrier[9], Lemur[10], etc; on the other hand. There are many Web search engines that access different databases, like: Google, Yahoo, AltaVista, etc.

[8] ftp://ftp.cs.cornell.edu/pub/smart/
[9] http://ir.dcs.gla.ac.uk/terrier/
[10] http://www.lemurproject.org/

It is very important to differentiate between collection fusion, where results from different data sets are merged, as shown in Figure 6; this fusion type is more suitable for the Web, since it is a very large database, and there is available many search engines that have access to different databases. There are many collection fusion systems where the main task is to fuse the results from different sources: MetaCrawler[11], Profusion[12], Search.com[13], etc. On another side, in data fusion the results from identical data sets are merged, as shown in Figure 7. The two fusion types are related but distinct: in collection fusion the goal is to cover different database, while in data fusion the goal is to improve the performance over any one of the systems being combined.

In our work, we restrict ourselves to the study of data fusion and we call it Model Fusion, since we fuse the results of different systems which are based on different IR models.

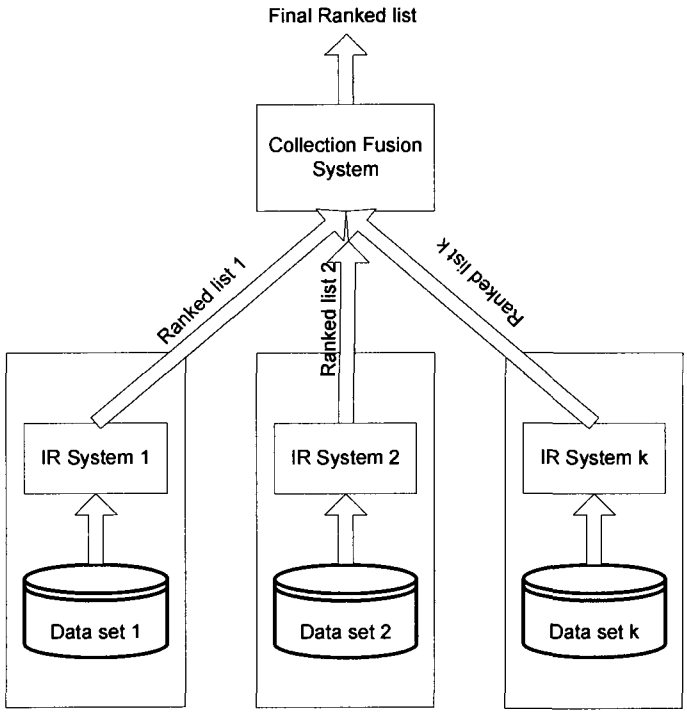

Figure 6: The structure of a collection fusion system which combines the results of k IR systems, where each IR system has access to different data set.

[11] www.metacrawler.com
[12] www.profusion.com
[13] www.search.com

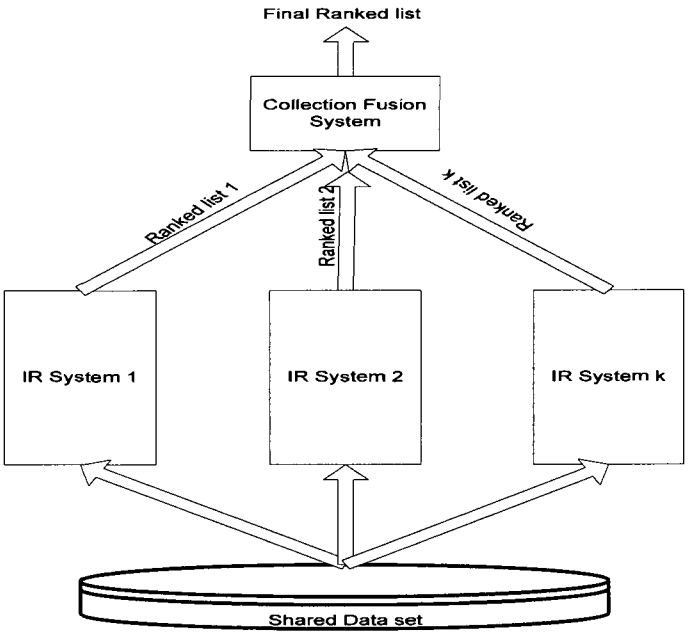

Figure 7: The structure of data fusion system which combines the results of k IR system, where each IR system has access to the same data set.

The data fusion approach

Several researchers showed improving effectiveness by combining the results of different retrieval strategies and different query representations. The hope is that each retrieval strategy will retrieve different sets of relevant documents, and combining the results will lead to a better result than any of the individual strategies. This is somewhat intuitive and many meta-search engines on the Web were developed in the hope of capitalizing on the notion that fusion will result in improved effectiveness. However, it is difficult to assess the effectiveness of meta-search engines because they are not typically run against a standard document collection with know relevant results.

Fusion of retrieval results from different models to improve the retrieval performance has been reported in works such as [13-16, 108, 109]. Retrieval results from different systems [108] or retrieval results using different document representations [15] were fused together for performance improvement.

Initial work on fusion was done by Fox with the TREC-1 collection [110]. He experimented with five weighting schemes based on vector space model (5 different systems). He took the top 200 results from each system, and then sorted the final results according to the initial performance, based

on the precision value at recall 0.0. If system A performed better than system B, then the results of system A were ranked before the results of system B. He concluded that his merging algorithm was not as good as the best individual run.

In 1993, with TREC-2 data, Fox and Shaw proposed six methods for combining five systems runs, based on the similarity values of documents to each query for each of the runs[12, 108]. Their proposed methods are shown in Table 6. The best improvement was achieved by using *CombSUM*, which is the summation of the set of similarity values of the documents for each system's run, as in the following formula:

$$CombSUM = \sum_{i \in IR\ schemes} score_i \qquad (2.70)$$

where $score_i$ is the similarity score of the document to the query for the weighting scheme i which retrieved this document.

Another effective fusion formula proposed by[108] is called *CombMNZ*, which sums up all the scores of a document multiplied by the number of non-zero scores of the document, as in the following formula:

$$CombMNZ = \sum_{i \in IR\ schemes} score_i * n \qquad (2.71)$$

where $score_i$ is the similarity score of the document for the weighting scheme i which retrieved this document, and n is the number of non-zero scores of the document.

Since there are different weighting schemes from different systems, these schemes will generate different ranges of similarity scores, so it is necessary to normalize the similarity scores of the documents. Lee [109] proposed a normalization method by utilizing the maximum and minimum scores for each weighting scheme as defined by the following formula .

$$NormalizedScore = \frac{score - MinScore}{MaxScore - MinScore} \qquad (2.72)$$

When training data is available, many researchers experimented with updated versions of *CombSUM and CombMNZ*, where a weight is assigned to each retrieval strategy according to performance on the training data. Then, they applied the determined fusion formula to the test data. These fusion methods are called *WCombSUM* and *WCombMNZ*, as shown in the following formulas:

$$WCombSUM = \sum_{i \in IR\ schemes} W_{ik} * NormalizedScore_i \quad (2.73)$$

$$WCombMNZ = \sum_{i \in IR\ schemes} W_{ik} * Normalizedscore_i * n \quad (2.74)$$

where W_{ik} is a pre-calculated weight associated with each weighting scheme's results in the cluster k, n is the number of non-zero scores of the document, and the NormalizedScorei is calculated as described before.

In 1994, Shaw and Fox found that combination of the runs form the same system with different queries types, i.e., long and short queries, with vector space model did not achieve improvements and sometimes degraded the performance. However, they achieved improvements over individual runs when merging two different model [108].

Several researchers attempted to characterize what makes result sets good candidates for effective fusion. In 1995, Lee classified the types of documents depending on the tf-vector length and the number of topics contained in the documents, and described the properties of weighting schemes such as cosine normalization and maximum normalization. Then, he explained that different types of documents may be retrieved by different properties of weighting schemes. He also showed through experiments that significant improvements can be obtained by combining the two retrieval runs in which one performs cosine normalization and the other does not [109].

Table 6: Combining formulas proposed by Fox and Shaw.

Name	**Combined Similarity**
CombMAX	Maximum of the individual measures.
CombMIN	Minimum of the individual measures.
CombMED	Median of the individual measures.
CombSUM	Sum of the individual measures.
CombANZ	Average of the individual measures. $CombANZ = CombSUM / Number\ of\ Nonzero\ Similarities$
CombMNZ	$CombMNZ = CombSUM * Number\ of\ Nonzero\ Similarities$

In 1997, Lee [111] found that a measure of overlap between results is an important factor. He analyzed the overlap values of results sets from six different participants in TREC-3. The overlap ratios of relevant and non-relevant documents ($R_{overlap}$ and $N_{overlap}$) are calculated as follows:

$$R_{overlap} = \frac{R_{common} \times 2}{R_1 + R_2} \qquad (2.75)$$

$$N_{overlap} = \frac{N_{common} \times 2}{N_1 + N_2} \qquad (2.76)$$

where R_{common} is the number of common relevant documents, R_1 is the numbers of relevant documents in run1, R_2 is the numbers of relevant documents in run2, N_{common} is the number of common non-relevant documents, N_1 is the number of non-relevant documents in run1, and N_2 is the number of non-relevant documents in run2.

Lee found that low overlap in non-relevant and high overlap in relevant documents is critical to improving effectiveness. The best improvement was achieved by the *CombMNZ* formula. He investigated the effect of using ranks instead of similarity on retrieval effectiveness, and found that using ranks gives better retrieval effectiveness, if the runs in the combination generate quite different rank-similarity curves.

In 1998, Vogt and Cottrell [112] suggests several characteristics for effective fusion. They tested numerous linear combinations (*CombSUM*) of several results from TREC-5, examining 36,600 results pairs. A linear regression of several potential indicators was performed to determine the potential for result sets to be fused. Thirteen factors, including measures of individual inputs, such as average precision/recall and some pairwise factors, such as overlap and unique document counts were considered. Vogt and Cottrell concluded five characteristics for effective fusion:

- At least one result has high precision/recall.
- High overlap of relevant documents.
- Low overlap of non-relevant documents.
- Similar distribution of relevance scores.
- Each retrieval system ranks relevant documents differently.

Chapter 3. Retrieval Tasks and Collection Description

In this chapter, descriptions of the retrieval tasks are first given. These include the monolingual and cross-language retrieval tasks. Details of the speech collections used in the experiments are then presented. The document collections is a part of the oral testimonies collected by the USC Shoah Foundation Institute for Visual History and Education (VHI) for which some Automatic Speech Recognition (ASR) transcriptions are available. Based on the characteristics of this collection, the retrieval tasks are then formulated appropriately for evaluation of the proposed retrieval methods. In addition, the description of the performance measures will be presented.

3.1. Retrieval tasks

In this work, the proposed retrieval approaches are investigated for both monolingual and cross-language speech retrieval (CL-SR) tasks. Details of these two retrieval tasks are given in the following subsections.

3.1.1 Monolingual Retrieval Task

In monolingual SR, the user-specified queries and the transcribed speech segments are in the same language. For our monolingual SR experiments, textual queries in English are used for searching spoken documents in English. The SDR task is performed by automatic transcription of the spoken documents and the transcriptions are then searched with the English textual queries. The retrieval output is a list of spoken documents (transcribed speech segments) ranked in order of their retrieval scores.

3.1.2 Cross-Language Retrieval Task

For a cross-language retrieval task, the user-specified queries and the spoken documents are in different languages. In CL-SR, there is an additional cross-language translation stage required to transform the queries into the same language as the documents. In our CL-SR experiments, tex-

tual queries in French, Spanish, and German are used for searching spoken documents in English. Details of the CL-SDR experiments will be given in Chapter 5.

3.2. Collection

The CLEF Cross-Language Speech Retrieval (CL-SR) task provides a standard test collection – the MALACH (Multilingual Access to Large Spoken Archives) – to identify and retrieve topically coherent-segments of English interviews in a known-boundary condition.

The document collection for the CL-SR task is a part of the oral testimonies collected by the USC Shoah Foundation Institute for Visual History and Education for which some Automatic Speech Recognition transcriptions are available [11]. The data is conversional spontaneous speech lacking clear topic boundaries.

Investigating the retrieval of spontaneous speech is more challenging than the retrieval of news broadcast like the one in TREC-SDR track for the following reasons [11, 113]:

- The lack of clear topic boundaries in conversational speech;
- The Large-Vocabulary Continuous Speech Recognition (referred to here generically as Automatic Speech Recognition, or ASR) techniques on which fully-automatic content-based search systems are based;
- In conversational spontaneous speech, many of the important words are not articulated between participants while expressing an opinion, developing an idea or clarifying some points, if they are not native speakers; many of the obvious search words are not present in the speech, so this problem would be significant if the conversations were not accurately transcribed.
- Many characteristics of the spontaneous speech in CLEF CL_SR collection affect the correctness of the Automatic Speech Recognition System used to transcribe the recordings: heavy accents, age-related co-articulations, speaker and language switching, and emotional speech. Transcription is challenging even for skilled annotators and they typically required 8 to 12 hours to transcribe a single hour of an English interview.
- The corpus consists of speech from elderly people; the age of the interviewees ranges from 56 years to 90 years.

- A large number of the words uttered in this corpus are foreign words or sequences of words spoken in a foreign language, plus unfamiliar names and places.
- Multiple languages encountered during a single interview.

The CL-SR track at CLEF was run for three consecutive years 2005, 2006, and 2007. The following subsections will present the two versions of the collection.

3.2.1 CLEF-2005 CL-SR collection

The CLEF-2005 CL-SR collection includes 8,104 manually-determined topically-coherent segments from 272 interviews with Holocaust survivors, witnesses and rescuers, totaling 589 hours of speech. Two ASR transcripts are available for this data (ASRTEXT2003A and ASRTEXT2004A), The transcripts were provided by IBM Research in 2004 for which a mean word error rate (WER) of 38% was computed on held out data [113]. Additional, meta-data fields for each segment include: two sets of 20 automatically assigned thesaurus terms from different kNN classifiers (AUTOKEYWORD2004A1 and AUTOKEYWORD2004A2), an average of 5 manually-assigned thesaurus terms (MANUALKEYWORD), and a 3-sentence summary written by a subject matter expert (SUMMARY). Figure 8 shows the structure of the segment. A set of 38 training topics and 25 test topics were generated in English from actual user requests. Topics were structured into title (T), description (D) and narrative (N) fields, which correspond roughly to a 2-3 word Web query, what someone might first say to a librarian, and what that librarian might ultimately understand after a brief reference interview, respectively. Figure 9 shows an example of a topic. To support CL-SR experiments, the topics were re-expressed in Czech, German, French, and Spanish by native speakers in a manner reflecting the way questions would be posed in those languages. Relevance judgments were manually generated using an augmented interactive search-guided procedure and purposive sampling designed to identify additional relevant segments. See [11] and [113] for details.

```
<DOC>
<DOCNO>VHF[IntCode]-[SegId].[SequenceNum]</DOCNO>
<INTERVIEWDATA>Interviewee name(s) and birthdate</INTERVIEWDATA>
<NAME>Full name of every person mentioned</NAME>
<MANUALKEYWORD>Thesaurus keywords assigned to the seg-
ment</MANUALKEYWORD>
<SUMMARY>3-sentence segment summary</SUMMARY>
<ASRTEXT2003A>ASR transcript produced in 2003</ASRTEXT2003A>
<ASRTEXT2004A>ASR transcript produced in 2004</ASRTEXT2004A>
<AUTOKEYWORD2004A1>Thesaurus keywords from a kNN classi-
fier</AUTOKEYWORD2004A1>
<AUTOKEYWORD2004A2>Thesaurus keywords from a second kNN classi-
fier</AUTOKEYWORD2004A2>
</DOC>
```

Figure 8 Document (segments) structure in CL-SR test collection.

```
<top>
<num> 1148
<title> Jewish resistance in Europe
<desc> Provide testimonies or describe actions of Jewish resistance in Europe before and
during the war.
<narr> The relevant material should describe actions of only- or mostly Jewish resis-
tancein Europe. Both individual and group-based actions are relevant. Type of actions
may include survival (fleeing, hiding, saving children), testifying (alerting the outside
world, writing, hiding testimonies), fighting (partisans, uprising, political security) In-
formation about undifferentiated resistance groups is not relevant.
</top>
```

Figure 9 Example topic.

3.2.2 CLEF 2006/2007 CL-SR collection

The segments used for the CLEF 2006 CL-SR task were identical to those used for CLEF 2005. Two new fields contain ASR transcripts of higher accuracy than those that were available in 2005 (ASRTEXT2006A and ASRTEXT2006B). The ASRTEXT2006A field contains a transcript which has a mean word error rate of 25% evaluated on held-out data. Only 7,378 segments have text in this field. For the remaining 726 segments, no ASR output was available from the 2006A system at the time the collection was distributed. The ASRTEXT2006B field seeks to avoid this no-content condition by including content identical to the ASRTEXT2006A field when available and content identical to the ASRTEXT2004A field otherwise.

The collection contains 105 search topics: 63 search topics from 2005 as training topic and 42 topics as a candidate test topics. 9 topics were rejected because they had either too few known relevant segments (fewer than 5) or too high density of known relevant segments among the available judgments; so only 33 topics were used for testing. The relevance assessments were created by experts: 28,223 binary (relevant or not) judgments for the 33 topics were created; among the judged documents 2,450 (8.6%) were relevant. See [6] for more details about the collection.

The collection for 2007 was the same as 2006 , 8,104 segments, 63 training topics, and 33 test topics [7].

3.3. Performance Measures

It is very important to evaluate each retrieval technique. Evaluation is based on testing and comparing IR systems in which different research hypotheses have been stated. The systems are tested using performance measures like precision or recall. The performance of an IR system for different queries can be quite different. To get a robust idea about the average performance of a system, the performance is measured over a set of queries in order to compute an average performance which can be represented by Mean Average Precision (MAP). The variation in retrieval performance across different queries is much larger than the variation of the averaged performance measure across systems (different hypotheses) because some queries are much harder than others for all systems. We can assess the significance of the conclusions with statistical tests (hypothesis testing techniques), which are able to detect consistent and significant performance differences between systems despite the noise introduced by query variation; one of hypothesis testing techniques used in IR is the Wilcoxon signed rank test which was used in all the experiments we have reported, with $p<0.05$, unless a different p value is stated.

In information retrieval, the performance measures are based on precision and recall.

Precision refers to how much useful information there is in the retrieval results. It is generally defined as the proportion of retrieved documents that are relevant. It can be calculated using the following formula:

$$\text{Precision} = \frac{number\ of\ relevant\ documents\ retrieved}{number\ of\ \ documents\ retrieved} \qquad (3.1)$$

Recall evaluates a retrieval system by reporting the proportion of relevant documents that are retrieved. It can be calculated using the following formula:

$$\text{Recall} = \frac{number\ of\ relevant\ documents\ retrieved}{number\ of\ \ relevant\ documents} \qquad (3.2)$$

Recall gives an indication on the amount of information made available to the user with respect to all the useful information. However, evaluation using recall requires additional a priori knowledge of the amount of relevant documents, as well as their identity in the collection. Sometimes, this information may not be available, such as it is the case in Web search.

For example, suppose that the relevant documents for a given query are (d1, d3, d5, d8, d9, d11, d13, d14, d17, d20); We run an information retrieval system which retrieved a ranked list of 15 documents (**d1**, d4, d7, **d20**, **d14**, d6, d15, d2, **d11**, d19, **d5**, d10, d12, d16, d18) for the given query; 5 of the retrieved documents are relevant and are indicated in bold font. According to the previous definition of precision and recall, the precision for the system is 0.33 (five out of 15), while the recall is 0.50 (five out of 15).

Evaluation of retrieval performance is usually based on both precision and recall. This is because a system can return as many documents to the user as possible in order to get a high recall. However, by returning too many documents, the system will become less useful since the user will then have to choose from a large pool of possible results. In the extreme case, a recall of one can be achieved by returning all documents. However, this trivial solution is useless because the retrieval process has not extracted any useful information for the user and there will be a low retrieval precision. For these reasons, retrieval performances are sometimes evaluated by averaging the precision values over all recall levels.

Precision - recall curves

Precision-recall curves present the performance of a retrieval run in a graph where precision is plotted as a function of recall (PR curve). The basis for the computation for a PR curve is formed by the relevance judgments and the ranked document lists produced by the IR system for each query. It is easy to compute recall and precision for each rank in the list. It is not so easy however to compute the average precision as a function of recall across topics, since each topic has a different number of relevant documents. One way to average precision values over a set of queries is to compute interpolated precision values at fixed points of recall. A standard adopted interpolation algorithm is the one implemented in trec_eval[14]. At each fixed recall point the interpolated precision is defined as the maximum of the precision at fixed recall points greater than or equal to the recall value in query.

[14] http://trec.nist.gov/trec_eval/

$$\text{precision}(i) = \max(\text{precision}(j)) \quad \textit{where} \quad j \geq i$$

The interpolated data can be used to compute precision at eleven standard points: 0, 0.1, 0.2, ... 0.9, 1.0. [54] gives a detailed account of this procedure. Figure 10 gives an example of such a PR curve. Interpolation thus forms the basis for PR curves. One can also average the precision over the 11 standard points of recall: average 11-point interpolated precision. This precision measure is not recommended, because it is strictly based on interpolated data. A method of averaging which is more faithful to the actual data is Mean Average Precision, also referred to as average un-interpolated precision.

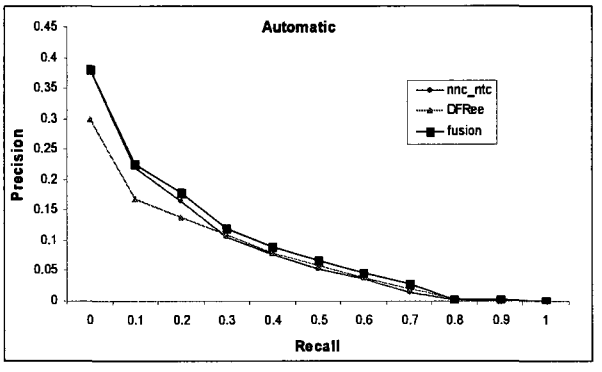

Figure 10 Example of a PR curve.

Mean Average Precision

In order to have one number that reflects both precision and recall, the average precision is used. The average precision for a certain query and a certain system can be computed by identifying the rank number n of each relevant document in a retrieval run. The corresponding precision is defined as the number of relevant documents found in the ranks equal or higher than the respective rank r divided by n. Relevant documents which are not retrieved receive a precision of zero. The average precision for a certain query is defined as the average value of the precision over all relevant documents. The Mean Average Precision (MAP) can be calculated by averaging the average precision over all queries using the following formula.

$$MAP = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{1}^{N_j}\text{precision}(d_{ij}) \qquad (3.3)$$

$$\text{precision}(d_{ij}) = \begin{cases} \dfrac{r_{n_i}}{n_i} & \textit{if } d_{ij} \textit{ retrieved} \\ 0 & \text{Otherwise} \end{cases} \qquad (3.4)$$

where n_i denotes the rank of the document d_{ij} which has been retrieved and is relevant for query j, r_{n_i} is the number of relevant documents found between ranks 1 and i ; N_j is the total number of relevant documents of query j, and M is the total number of queries. Using the same data from the previous example, the average precision is calculated in table 7.

Table 7: Calculation of Average Precision for one query.

The rank for document i (n_i)	**Number of relevant documents**	**Documents found relevant**	**Precision**
1	1	d_1	1/1 = 1
4	2	d_{20}	2/4 = 0.5
5	3	d_{14}	3/5 = 0.6
9	4	d_{11}	4/9 = 0.4444
11	5	d_5	5/11 = 0.4545
Not retrieved		d_3	0
Not retrieved		d_8	0
Not retrieved		d_9	0
Not retrieved		d_{13}	0
Not retrieved		d_{17}	0
Average Precision			0.2999

R-Precision

R-precision is the precision at R where R is the number of relevant documents in the collection for the query. An R-precision of 1.0 is equivalent to perfect relevance ranking and perfect recall. However, a typical value of R-precision which is far below 1.0 does not indicate the actual value of recall (since some of the relevant documents may be present in the hitlist beyond point R)

Chapter 4. Related Work from CLEF and TREC

4.1. Related Work from CLEF

4.1.1 Description of the Systems that Participated in CLEF-CLSR 2005

Seven groups submitted runs to this track [5]. A brief description of their experiments is given in the following subsections.

University of Alicante

University of Alicante [114] used a passage retrieval system for their experiments. Passages in such systems are usually composed of a fixed number of sentences. However, the format of spoken documents does not allow this supposition. In this case, the documents are composed by a fixed number of contiguous sets of words. Their experimental system applied heuristics to the representation of the topics in the way of logic forms. The logic form of a topic (or sentence) was derived through the processing of the dependency tree between the words of the sentence.

Dublin City University

Dublin City University's system [115] was based on a version of the Okapi probabilistic model. Queries were expanded using pseudo relevance feedback (PRF). Expansion terms were selected from summaries of the top 5 most relevant documents (assumed by the system), where "sentences" in the ASR transcript were derived based on sequential word clusters. All terms within the chosen sentences were then ranked and the top 20 ranking terms were selected as expansion terms. They have explored the effect of various combinations of the ASR transcripts and the meta-data provided with the collection.

University of Maryland

The University of Maryland [116] used the InQuery information retrieval system from the University of Massachusetts. They have tried different techniques such as document expansion with manually created meta-data (thesaurus keywords and segment summaries) from a large side collection,

query refinement with pseudo-relevance feedback, and keyword expansion with thesaurus synonyms. Their experiments showed that a combination of document expansion using a side collection and query expansion using the collection being searched could improve speech retrieval effectiveness and that tuning the expansion parameters on the training topics yielded near-optimal improvements on the evaluation topics.

National University of Distance Education (UNED)

UNED [117] tried several strategies to clean up the automatic transcriptions. They erased all duplicate words and joined the characters that form spelled words like "l i e b b a c h a r d" into the whole word (i.e., "liebbachard"). Using this cleaned collection they tried a monolingual trigrams approach by splitting words into 3-grams of characters. They also tried to clean the documents, erasing the less informative words using two different approaches: morphological analysis and part-of-speech tagging, where all words labeled either as a noun, an adjective or a verb were included into collection. They have tried to improve the query structure by using an entity recognizer to identify possible named entities in the topics. They have studied the pseudo-relevance feedback approach using manual keywords. Their experiments showed that using a shallow entity recognizer to identify named entities improved the retrieval. There was no significant differences between the cleaning methods, except for the 3-gram which performed worst. Pseudo-relevance feedback using manually-generated keywords turned out to be the best option to improve retrieval performance.

University of Pittsburgh

The University of Pittsburgh [118] explored data fusion techniques for integrating the manually-generated meta-data information with the ASR transcripts. They have explored a weighted CombMNZ model with different weight ratios and multiple iterations. Their initial results indicate that a simple unweighted combination method, that has been demonstrated to be useful in written retrieval environments, generated a 38% relative decrease in retrieval effectiveness. They have shown that multiple-iteration fusion with the weighted CombMNZ improved the retrieval. The weights were selected manually in an ad-hoc manner. Their basic runs was produced using Indri 1.0, which is a collaboration effort between the University of Massachusetts and Carnegie Mellon University. Its retrieval model is a combination of language model and inference network model.

University of Waterloo

The University of Waterloo system [119] was based on Okapi BM25. They have explored several query formulation and expansion techniques, including the use of phonetic n-grams and feedback query expansion over a topic-specific external corpus crawled from the Web.

For feedback purposes, they augmented the CLEF 2005 CL-SR corpus with the 2.5GB corpus of Web data, generated by a topic-focused crawl, seeded from 17 sites dedicated to the holocaust. Each query was first executed against this augmented corpus. Terms were extracted from the top results and added to the initial query, which was then executed against the CL-SR corpus.

As an alternative to stemming, many runs were based on phoneme 4-grams. For these runs, NIST's text-to-phone tool was applied to translate the words in the corpus into phoneme sequences, which were then split into 4-grams and indexed. Queries were preprocessed in a similar fashion before execution.

Their experiments results have showed that feedback produced only a modest improvement and the phonetic n-grams generally harmed performance.

4.1.2 Description of the Systems that Participated in CLEF-CLSR 2006

Six groups submitted runs to this track [6]. A brief description of their experiments is presented in the following subsections.

University of Alicante (UA)

The University of Alicante system [120] used two stages: the first one consisted in increasing the weight of some topic terms by applying a set of rules that are based on the representation of the topics by means of logic forms which are based on the analysis of syntactic dependencies in the topic descriptions and in the automatically-generated portion of the collection; and the second one consisted in only applying IR-n system -which uses the Okapi similarity measure- to the collection, to produce overlapping passages.

Dublin City University (DCU)

Dublin City University used a system based on the Okapi retrieval model [121, 122]. Their experiments explored the combination of multiple segment fields using the BM25F variant of Okapi weights [123]. The combination process was executed during the indexing phase. The BM25F combination approach uses a simple weighted summation of the multiple fields of the documents to form

a single field for each document. The importance of each document field for retrieval is determined empirically in separate runs for each field; the count of each term appearing in each field is multiplied by a scalar constant representing the importance of this field. The components of all fields are then summed to form the overall single-field document representation for indexing. Once the fields have been combined in a weighted sum, standard single field IR methods were applied. Their system was also used to explore the use of a field-based method for term selection in query expansion with pseudo-relevance feedback.

University of Maryland (UMD)

The University of Maryland team used the InQuery system to run their experiments [124]. They tried different pre-indexing combinations of that data. They combined the ASR text generated with the 2004 system and the ASR text generated with the 2006 system, and we compared those results with results obtained by indexing each alone. Their motivation was that the two ASR systems would produce different errors, and that combining their results might therefore yield better retrieval effectiveness which was confirmed by their experiments. Three CLIR experiments were conducted using French topics, but an apparent domain mismatch between the source of the translation probabilities and the CLEF CL-SR test collection resulted in relatively poor MAP for all cross-language runs.

National University of Distance Education (UNED)

The UNED team used the same as that for their participation in the CLEF 2005 CL-SR tasks[6, 117]. They have focused their experiments to compare between the 2006 ASR with manually generated summaries and manually assigned keywords. A CLIR experiment was performed using Spanish queries with the 2006 ASR.

University of Twente (UT)

The University of Twente used a locally developed XML retrieval system to flexibly index the collection in a way that permitted experiments to be run with different combinations of fields without reindexing [125]. Their experiments showed the combination of manual and automatically generated annotations will be beneficial to improve the retrieval. Cross-language experiments were conducted using Dutch topics, both with ASR and with manually created meta-data.

4.1.3 Description of the Systems that Participated in CLEF-CLSR 2007

Six groups submitted runs [7]; a brief description of their experiments is given in the following subsections.

Brown University (BLLIP)

The Brown University system was based on the language model (LM) for information retrieval [126]. They extended the basic Dirichlet-smoothed unigram IR model into a bigram IR model. They presented two smoothing-based extensions in the LM paradigm. The bigram and unigram models were mixed using a weighted mixture over all documents. In order to accommodate the sparse data problems in the case of small collections, their model attempted to mix the test collection with two larger text corpora. Their experiments showed that their model with pseudo relevance feedback had an improvement over the simple unigram model.

Dublin City University (DCU)

Dublin City University system was based on the Okapi model (BM25F formula) and used a combination of multiple fields with summary-based pseudo-relevance feedback [127]. Their pseudo-relevance feedback method showed a statistically-significant improvement on manual data, but not on automatic data. Their cross language experiments were based on translating the non-English topics into English using Yahoo! BabelFish[15] free online translation service and using domain-specific translation lexicons gathered automatically from Wikipedia.

University of Amsterdam (UVA)

The University of Amsterdam [128]used the Indri engine from the Lemur retrieval toolkit[16] for their experiments. They explored the use of character n-gram tokenization. Their experiments showed that 4-grams provided the best retrieval effectiveness for cross language experiments, but this was not the case for monolingual experiments.

They examined the effects of combining manual and automatically-generated data. They combined the fields using the Indri engine where different weights were assigned to different fields. Their experiments showed that the combination of manual and automatically-generated data improved the retrieval.

[15] http://babelfish.yahoo.com/
[16] http://www.lemurproject.org/

University of Chicago (UC)

The University of Chicago system [129] was based on the InQuery information retrieval system [130]. Their experiments focused on the contribution of automatically-assigned thesaurus terms to retrieval effectiveness and the utility of different query translation strategies. For cross-language retrieval, they used two query translation strategies: machine translation using the tool provided by Google, and dictionary-based translation. Their experiments showed that the machine translation tool was better than the dictionary-based translation and that assigning automatic keywords improve the retrieval.

University of Ja´en (SINAI)

The University of Ja´en team [131] conducted their experiments with the Lemur retrieval information system. They investigated the effect of selecting of different fields on the retrieval effectiveness. An information gain measure was computed for each field in the segment to select the best field in the document collection. The fields with higher information gain values were selected to compose the final collection. Their experiments were conducted with Lemur using the KL divergence. For cross-language experiments, they combined the output of different online machine translation systems based on heuristics.

4.2. Related Work from TREC

4.2.1 Description of the Systems that Participated in TREC6-SDR 1997

The first SDR evaluation was organized in 1997 for TREC-6 [3]. This evaluation implemented a "known-item" task in which a particular relevant document was to be retrieved for each set of queries over a 50-hour collection of radio and television news broadcasts. This collection is very different from the CLEF-CLSR collections because that speech conditions are much better for broadcast news, therefore the word error rate is reduced.

Three retrieval conditions were implemented to examine the effect of the speech recognition performance on the retrieval performance:

1. Reference - retrieval using human-generated speech transcripts.

2. Baseline - retrieval using the transcripts generated by the IBM speech recognizer, with a 50% estimated word error rate.

3. Speech - retrieval using the recordings of the broadcasts themselves, requiring both speech recognition and information retrieval technologies.

Thirteen sites participated in the evaluation, eight of which implemented the Speech retrieval condition using their own or another team's speech recognition system. Brief descriptions of their systems and experiments are given in the following subsections. The University of Massachusetts retrieval system yielded the best performance for all three conditions. The UMass system achieved a retrieval rate of 78.7% for the Reference Retrieval condition, and 63.8% for the Baseline Retrieval condition. For the full SDR condition, UMass used a transcript produced with the Dragon Naturally-Speaking system, with a 35% word error rate, and achieved a 76.6% retrieval rate.

Figure 11 shows the results for all the systems that participated in the TREC6-SDR track. The first SDR evaluation successfully implemented an evaluation of SDR technology and showed that existing components and tools worked well on a known-item task with a small audio collection.

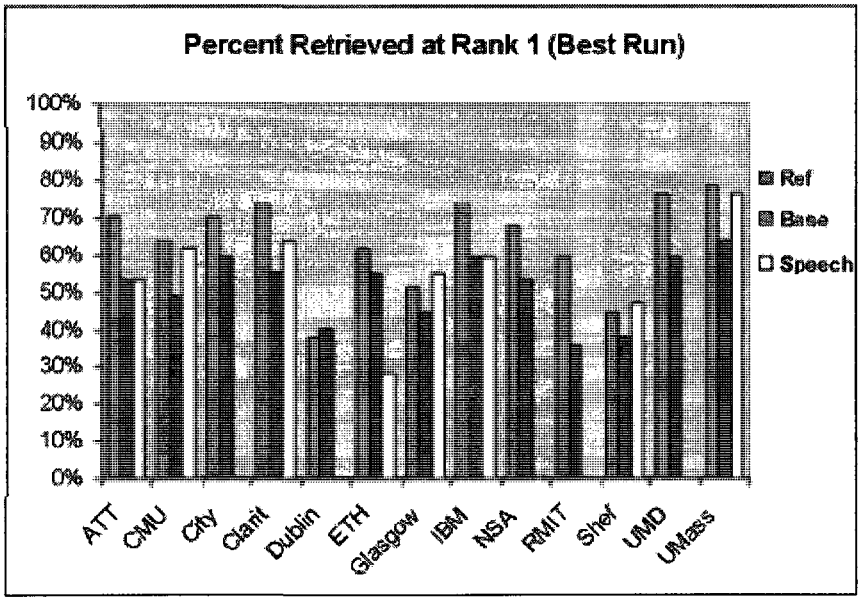

Figure 11: Results for all participating systems in TREC6-SDR.

AT&T Lab (ATT)

The AT&T lab [132] investigated how higher word-recall – recognizing many of the spoken words – affects the retrieval effectiveness for speech documents, given that high word-recall comes at a cost of low word-precision – recognizing many words that were not actually spoken. They simulated a

high word-recall and a poor word-precision system by merging the output of several recognizers. Their information retrieval system was based on the SMART system [50], where the weighting schemes for the documents and queries are bnu and ltu, respectively. Experiments suggest that having higher word-recall does improve the retrieval effectiveness from speech.

Carnegie Mellon University (CMU)

The Carnegie Mellon University [133] team explored the effect of merging two speech recognition outputs with different word error rates, and compensating for missing words which are critical to the retrieval by adding probable words from the speech recognizer's hypothesized N-best list. Their information retrieval system used the TF-IDF weighting scheme. Their experiments showed that combining the results of two independent recognition systems slightly boosted information retrieval results, and using the N-best list to augment the speech recognition output with likely words showed great promise, by reducing the difference between perfect text transcripts and speech recognizer-generated transcripts.

City University of London (City)

The City University of London [134] system was based on the Okapi information retrieval system. They compared search performance for un-weighted terms (UW) terms with only collection frequency weighting (CFW) and terms with full Okapi-style combined weighting (CW, aka BM25). Their research questions were whether relative performance for these strategies was the same for two recognition systems, and how the weighting formula in particular behaved in improving targeting of the known items and compensating for speech recognition deficiencies.

Dublin City University (Dublin)

Dublin City University's [135] team investigated two indexing strategies: triphones (concatenation of three phones where phones are taken from an alphabet of 41 possible phones) and full-words. Their system intended to remove the tags used for story boundaries, turned transcripts and topics into a phonetic representation using a phoneme dictionary, and then retrieved story identifiers based on the triphone match between topic and fixed-width windows of triphones in the transcripts. They also applied a weighting function to triphones as they occurred in story "windows" based on their offset within those windows. Their information retrieval system used the TF-IDF weighting scheme. Their experiments showed that the triphones work better than words for the baseline (speech recognition data), but it was not the case for reference data (manual transcribed data).

Swiss Federal Institute of Technology

The Swiss Federal Institute of Technology team [136] developed a speech recognition system for English; for the retrieval part, their system translated the queries and transcriptions to phonemic transcriptions. They have proposed a probabilistic approach that includes new document length and feature length normalizations in the weighting stage and a new compensation measure in the retrieval method to normalize the variability in expected Retrieval Status Values (RSV) compared to the actual RSV which arises across documents with different numbers of query features.

University of Glasgow (Glasgow)

University of Glasgow [137] used an information retrieval system called SIRE (System for Information Retrieval Experimentation) developed at Glasgow University. The weighting scheme in SIRE is based on PTF-IDF, where the PTF part was calculated based on the probabilities generated by the speech recognition system. Their experiments focused on merging the transcripts from two speech recognition systems to compensate for the errors that may result from one of the recognizers, and on comparing between two weighting schemes: TF-IDF and PTF-IDF. Performance of merging two recognizers was better than each individual recognizer, and the PTF-IDF weighting scheme gave better performance than the classical TF-IDF.

Royal Melbourne Institute of Technology (RMIT)

The Royal Melbourne Institute of Technology [138] used an information retrieval system based on the vector space model with a cosine similarity and TF-IDF weighting scheme. They have explored the use of phoneme sequences as matching units, instead of words. Phonemes were extracted from the speech tracks and triphones were created to perform retrieval. For comparison with the manual transcripts, the transcripts were also translated to phoneme sequence. The intuition behind using triphones it that they have a higher noise tolerance than words, and word boundaries become less important. Their experiments showed that using triphones improve the retrieval over full words.

U.S. Department of Defense

The U.S. Department of Defense team [139] built an information retrieval system which used a software called Semantic Forests which is based on an algorithm originally developed for labeling topics in text and transcribed speech [140]. Topic-labeling is not an IR task, so Semantic Forests were adapted for use in TREC. Semantic Forests report terms as found in its dictionary that are conceptually similar (synonymous); the hope was to improve the retrieval by adding these synonyms.

University of Sheffield (Shef)

The University of Sheffield Spoken Document Retrieval System [141] is based on the ABBOT Large Vocabulary Continuous Speech Recognition (LVCSR) system developed by Cambridge University, Sheffield University and SoftSound, and uses PRISE[17] for indexing and retrieval which was developed by National Institute of Standards and Technology (NIST). They have investigated the effect of a word error rate of 30-50% on retrieval performance by comparing their speech recognition system with the system developed by IBM. Their experiments showed that their system was better than the IBM system.

University of Massachusetts (UMass)

The University of Massachusetts team [17] used the Inquery information retrieval system and the Dragon speech recognition system for their experiments. Local Context Analysis (LCA) locates expansion terms in top ranked passages, uses phrases, as well as terms, for expansion features, and weights the features in a way intended to boost the expected value of features that regularly occur near the query terms. LCA was applied to two different collections: the test data and the Topic Detection and Tracking (TDT) corpus available via the Linguistic Data Consortium, which covered a similar time period as the test corpus. A comparison of the results using the Dragon system and the IBM system was conducted. Another comparison was between LCA computed from the test collection and from the TDT collection. The experiments showed that the Dragon system was better than the IBM system, and LCA from the test collection was better than from the TDT collection.

University of Maryland (UMD)

University of Maryland team [142] used an Inquery information retrieval system; their experiments focused on how the speech recognition output affects the retrieval by doing a comparison between the manually-transcribed text and the IBM speech recognition output.

4.2.2 Description of the Systems that Participated in TREC7-SDR 1998

The second SDR evaluation was implemented in 1998 for TREC-7 [4], using a subset of 100 hours of the Broadcast News Corpus collected between June 1997 and January 1998. A final filtered test set contained 2,866 stories with about 772,000 words and 23 topics were selected for evaluation. The primary retrieval metric for the SDR evaluation was Mean Average Precision over all topics. Three

[17] http://www-nlpir.nist.gov/ir_tools.html

retrieval conditions were implemented to examine the effect of recognition performance on retrieval performance:

1. Reference(R) - retrieval using human-generated transcripts.
2. Baseline (B1/B2) - Retrieval using two sets of speech-recognition-generated 1-best transcripts produced by NIST using the CMU SPHINX-III recognition system, with word error rates of 33.8% and 46.6%.
3. Speech (S1/S2) - retrieval using the recordings of the broadcasts themselves requiring both speech recognition and retrieval technologies.

Eleven sites participated in the evaluation, eight of which implemented the Speech retrieval condition using their own or another existing team speech recognition system. Brief description of their systems and experiments are given in the following subsections. The University of Massachusetts' retrieval system yielded the best performance for all three conditions (as shown in the Figure 12). The UMass system achieved the best Mean Average Precision. The UMass system achieved a MAP score for the retrieval using the human reference transcripts (R1) of 0.5668. The same system's retrieval for the moderate-error baseline recognizer (B1) transcripts achieved a MAP of 0.5063, and for the high-error baseline recognizer (B2) transcripts, a MAP of 0.4191. For the Speech input condition (S1) using their own team's recognizer (the Dragon system) at 29.5% word error rate, the UMass system achieved a MAP of 0.5075.

The conclusion of the SDR evaluation that there is a near-linear relationship between the recognition word error rate and the retrieval performance. Figure 12 shows the results for all participated sites in TREC7-SDR

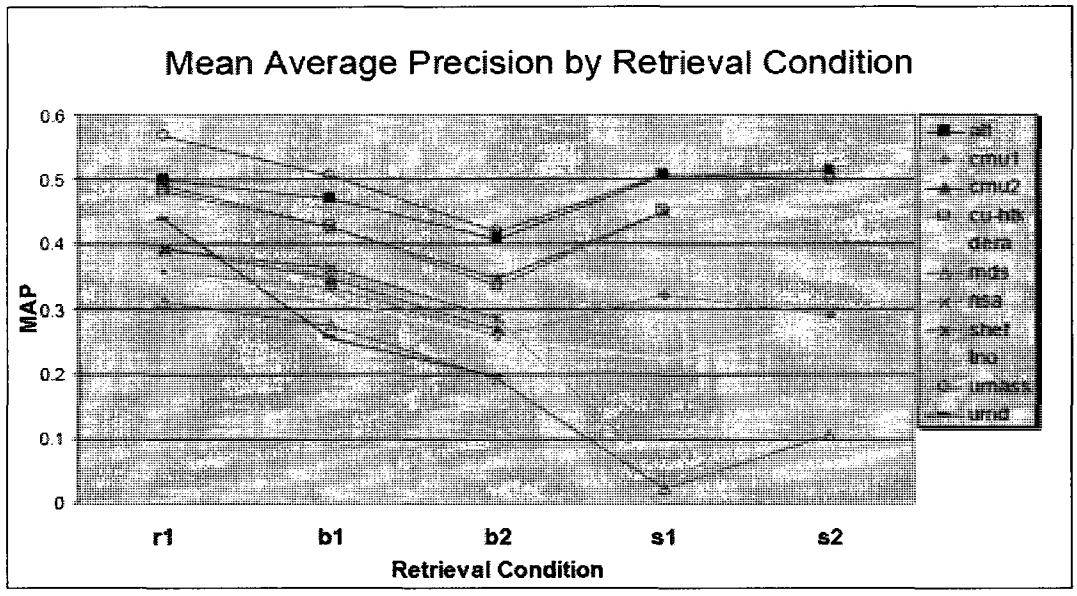

Figure 12 Results for all the systems that participated in TREC7-SDR.

AT&T Lab

The AT&T information retrieval system was based on the SMART system for retrieval [20]. They investigated three techniques: query expansion form the target collection, query expansion from the side collection, and document expansion from the side collection. The main idea when using a larger side collection for query and document expansion is that a larger text collection will have more relevant documents for the query and the retrieval methods will find more relevant documents in the top ten or twenty documents, thereby strengthening the assumption of relevance employed in the query expansion stage. Their experiments showed that document expansion reduces the performance gap between retrieval from perfect and automatic transcriptions. This confirms their belief that term recall plays a more important role in retrieval effectiveness for speech than term precision.

Carnegie Mellon University

Carnegie Mellon University's team [143] proposed two information retrieval systems, one based on probabilistic language model, and the other one based on vector space model with TF-IDF weighting that incorporates word error probability. They used simple confidence estimates for word probabilities based on speech recognition lattices. Their experiments showed that the language model retrieval engine worked better than TF-IDF.

Cambridge University, UK (CUHTK)

The Cambridge University team [144] used Okapi-based retrieval engine;. They investigated the effects of changing stop-lists by adding words that appeared specifically for the broadcast news data, such as "uh-huh" or "hmmm"; adding bad-spelling correctors such as "all right"/"alright" and "baby sit"/"baby-sit"/"babysit"; a stemming exceptions list and basic synonym mapping, including word-pair information; weighting query terms by their part-of-speech; and adding pre-search statistical expansion. These techniques have been shown to increase IR performance under certain circumstances, but the increases were small. Nevertheless, the combination of these devices led to an increase in average precision on the TREC-7 evaluation data of 2.74% on the reference and 2.27% on the automatic transcriptions.

Royal Melbourne Institute of Technology

Royal Melbourne Institute of Technology's system [145] was based on translating the documents into phoneme sequences. For the reference and baseline retrieval runs, the word-based documents and queries were translated into phonemes sequences using the CMU pronunciation dictionary, and the tri-grams and 4-grams sequences were combined for each document prior to retrieval. The queries used were the bounded 4-grams of the translated query; they were not allowed to cross word boundaries. The Okapi similarity function was used for retrieval. Their experiments showed that the retrieval using word-based documents is better than the phonetic n-gram combination retrieval.

U.S. Department of Defense

The U.S. Department of Defense system [146] was based on the Semantic Forests algorithm for labeling topics in text and transcribed speech. Semantic Forests used an electronic dictionary to make a tree for each word in a text document. The root of the tree is the word from the document; the first branches are the words in the definition of the root word; the next branches are the words in the definitions of the words in the first branches; and so on. The words in the trees are tagged by part of speech and given weights based on statistics gathered during training. Finally, the trees are merged into a scored list of words. The premise is that words in common between trees will be reinforced and they would represent "topics" present in the document. A part-of-speech tagger was added, to allow Semantic Forests to use this additional information.

DERA, UK (DERA)

The DERA team [147] investigated the effects of term expansion using a semantic network, such as Wordnet, on retrieval performance. They used the Okapi information retrieval system for their experiments. The query text is syntactically tagged and keywords are selected on the basis of their part-of-speech tags. The tags are used to extract a set of keywords and key phrases for the retrieval engine, by extracting words with tags that are likely to be discriminative. The output from the tagger is also processed to extract compound nouns and adjectival phrases. Finally, term expansion is performed using Wordnet. Their experiments showed that term expansion using Wordnet can occasionally be of great benefit, but more often lead to degradation in performance.

University of Sheffield

The University of Sheffield system [18] is based on the ABBOT speech recognition system and the thislIR text retrieval system which use Okapi term weighting scheme. They have investigated the use of multiple transcriptions and word graphs, the effect of simple query expansion algorithms from side collection, and the effect of varying standard IR parameters in the Okapi weighting formula. They concluded from their experiments that query expansion using a secondary collection derived from newswire data from a similar time period gives a consistent relative improvement in average precision of around 10%, and that it is more important to focus on retrieval strategy because it has a much greater effect than improving the speech recognizer.

TNO-TPD TU-Delft, Netherlands (TNO)

The TNO [148] Spoken Document Retrieval System is based on the ABBOT Large Vocabulary Continuous Speech Recognition (LVCSR) system developed by Cambridge University, Sheffield University and SoftSound, and used word spotting, the TNO Vector Space Engine, and fuzzy matching based on phoneme trigrams for indexing and retrieval. They investigated several approaches: Fuzzy matching on a phoneme representation of the database and phone lattice-based word spotting. They concluded that that phone based retrieval is a feasible and scaleable approach.

University of Massachusetts

The University of Massachusetts team [19] used an Inquery information retrieval system and the Dragon speech recognition system for their experiments. They concentrated their experiments to investigate query expansion using Local Context Analysis (LCA), where expansion terms in top ranked documents are selected, and then the final query is a weighted combination of the original

query and the expansion features. The expansion features was selected from the SDR collection, a corpus of AP newswire documents produced at the same time as the SDR audio broadcasts, or a combination of different versions of the SDR collection (each version produced by a different speech recognizer). They concluded that retrieval on audio data of the quality used in the SDR collection can be expected to be almost as good as retrieval on a hand-transcribed version.

University of Maryland

University of Maryland [149] used a modified version of PRISE for their experiments, in which some changes had been made to the numerical details of retrieval status value computation, but a comparison with the original system revealed no significant differences in the ranked output. The Porter stemmer, Okapi1 weights, and the Inquery stop word list were the only deviations from the default settings in the indexer.

4.2.3 Description of the Systems that Participated in TREC8-SDR 1999

The third SDR evaluation was implemented in 1999 for TREC-8 [1]. The final collection contained 557 hours of the Broadcast News Corpus collected between February 1, 1998 and June 30, 1998. A final filtered test set contained 21,754 stories, and 49 topics were selected for evaluation. The primary retrieval metric for the SDR is the MAP score.

In addition to the Reference, Baseline, Speech, and Cross Recognizer retrieval conditions used in TREC-7, an optional story boundaries unknown (SU) condition was added for TREC-8. This condition permitted sites to explore SDR where they had to operate on whole broadcasts with no knowledge of human-annotated topical boundaries. This condition more accurately represented the real SDR application challenge.

Ten sites participated in the evaluation, six of which implemented the Speech retrieval condition using their own or another speech recognition system. Brief description of their systems and experiments are described in the following subsections, and figure 13 shows the results for all the sites that participated in the TREC6-SDR track.

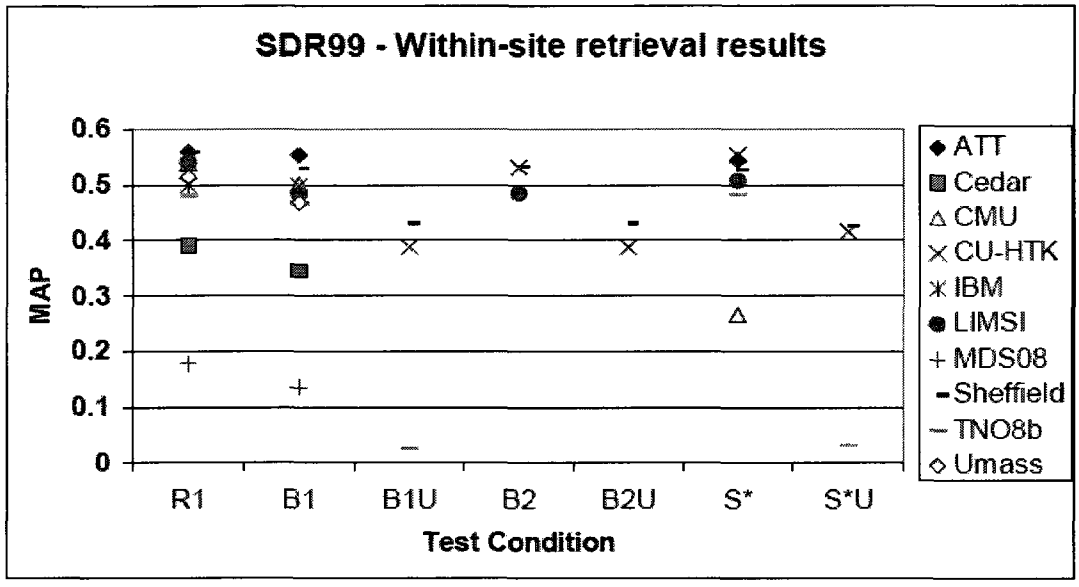

Figure 13: Results for all participating systems in TREC8-SDR.

AT&T Lab [ATT]

The AT&T information retrieval system was based on SMART system for retrieval[26, 150]. They investigated document expansion from side collection to compensate the transcription error on speech retrieval, and the effects of word error rate on speech retrieval. For document expansion, they used the NA News corpus and UPI news from the same period as the test collection. Their experiments showed that document expansion from a text collection (a side collection) closely related to the test collection yields to improvements in retrieval, and that the retrieval effectiveness is not very sensitive to the WER of the recognizers for reasonable recognition rates.

University of Cambridge [CU-HTK]

University of Cambridge retrieval engine [24] used an Okapi scheme with the following characteristics: traditional stop word lists and Porter stemming, weighting the query terms according to part-of-speech, a stemmer exceptions list, basic synonym mapping, parallel collection frequency weighting, both parallel and traditional blind relevance feedback, and document expansion using parallel blind relevance feedback. The combination of these techniques was shown to increase IR performance by 23%.

LIMSI [LIMSI]

LIMSI 's system [23] combined an adapted version of the LIMSI 1998 Hub-4E transcription system for speech recognition with an Okapi-based IR system. Another weighting scheme called Markovian weighting scheme has also been developed. They have investigated two query expansion techniques based on terms present in the retrieved documents on the same (Blind Relevance Feedback BRF) or another (Parallel Blind Relevance Feedback PBRF) data collection. Their experiments showed that the combination of the two query expansion techniques improved the retrieval, and very comparable results can be achieved using the two term-weighting schemes.

The Royal Melbourne Institute of Technology [MDS]

The Royal Melbourne Institute of Technology system [151] was based on passage retrieval using phoneme sequences. Documents and queries were translated into phonemes using a rule-based text-to-speech algorithm. A passage was created for each query term and approximate matches were computed within each document. These passages were combined using either cosine or Okapi-based weighting scores for each document, before similarity was computed for each query. Their experiments showed that phoneme-based retrieval of speech documents using word-based transcripts is not as effective as when words are used.

Sheffield University [SHEFFIELD]

The Sheffield University system [21] for retrieval was based on the Okapi term weighting scheme. They investigated local context analysis (LCA) for query expansion from a secondary corpus of documents from a similar domain, that does not contain recognition errors. The results show that information retrieval performance was not significantly affected by transcription errors, and the transcription errors were compensated by techniques such as query expansion.

IBM [IBM]

IBM system [152] was based on the Okapi weighting scheme. They investigated the translation model to reduce the impact of speech recognition errors on the performance of the information retrieval system, where the corpus of automatically-transcribed data is considered to be one language of a parallel corpus, and the corpus of manual transcriptions is considered to be the second language in a parallel corpus. Then retrieval of automatic transcriptions of broadcast news is considered to be a problem of cross-language information retrieval. They trained a statistical machine translation model to translate the documents from the language of automatically-transcribed data into the lan-

guage of manually-transcribed data. The test corpus was processed with this translation model, correcting some of the recognition errors, and establishing cleaner text features to be used by the information retrieval system.

Carnegie Mellon University [CMU]

The Carnegie Mellon University team [153] derived theoretic interpretations for the TF-IDF weighting scheme and pointed out a possible reason why multiplying tf directly with IDF causes the poor performance with the given interpretation. They proposed a word histogram method of integrating TF and IDF into one factor which is log(1/p') and p' is the probability that a document has at least TF occurrences of a particular word. Their experiments showed that the proposed weighting scheme is better than TF-IDF and (ln(tf)+1)*idf and failed when competing with (ln(ln(tf)+1)+1)*idf.

University of Massachusetts [UMASS]

The University of Massachusetts team [22] used the Inquery information retrieval system. They concentrated their experiments to investigate query expansion using Local Context Analysis (LCA), where expansion terms in top ranked documents are selected, and then final query is a weighted combination of the original query and the expansion features. The expansion features was selected from external collection produced at the same time as the SDR audio broadcasts. They concluded that retrieval on audio data of the quality used in the SDR collection can be expected to be almost as good as retrieval on a hand-transcribed version.

Twenty One Consortium [TNO].

The Twenty One Consortium system [154] was based on a probabilistic retrieval model based on statistical language models. They implemented Blind Relevance Feedback (BRF). The relevance feedback was applied on a larger secondary corpus: the TREC Ad Hoc corpus. Experiments showed that even though the corpus covers a different time periods, results with the secondary corpus were better than BRF on the SDR corpus only.

Chapter 5. Our First Approach to Retrieval From Automatic Speech Transcripts

5.1. Introduction

In this chapter, we present our first approach to retrieval from automatic transcripts of spontaneous speech. We compare various indexing schemes. For cross-language IR we combine multiple translations of the queries. We test the retrieval from phonetic transcription of the documents. We propose two relevance feedback models. We compare the retrieval from automatic transcripts, manual summaries and keywords, and combinations.

5.2. Comparison of Indexing Schemes

The first step in our work was to investigate different weighting schemes based on the vector space model generated from SMART system. The spontaneous speech collection was tested with many different weighting schemes for indexing the collection and the queries, using SMART system. We generated weighted term vectors for the document collection. SMART preprocessed the documents by tokenizing the text into words, removing common words that appear on its stop-list, and performing stemming on the remaining words to derive a set of terms. When the IR server executes a user query, the query terms are also converted into weighted term vectors. The weighting schemes are combinations of term frequency, collection frequency, and length normalization components as discussed in Section 2.4.2. There are 3600 possible combinations of weighting schemes: 60 schemes (5 x 4 x 3) for documents and 60 for queries. We tried 240 combinations and we present in Table 7 the results for 15 combinations (the best ones, plus some other ones to show the diversity of the results). One scheme in particular (the lnn.ntn weighting scheme) was used in our submissions for CLEF 2005, because it proved to have much better performance than other combinations. For weighting document terms, we used term frequency normalized by taking the logarithmic function for the terms frequency, and neither collection frequency weighting, nor cosine normalization were applied. For queries, we used non-normalized term frequency and inverse document frequency

weighting. Appendix A shows the formulas for all the weighting schemes that we used for documents or queries. Vector inner-product similarity computation is then used to rank documents in decreasing order of their similarity to the user query. For more detailed about these weighting and the similarity measure, see Section 2.4.2.

Table 8 presents results for various weighting schemes for document (before the dot in the standard notation) and for topics (after the dot). The lnn.ntn is still the best, but there are a few other weighting schemes that achieve similar performance. Some of the weighting schemes perform best when indexing all the fields of the queries (TDN = title, description, and narrative), some on TD (title and description), and some on title only (T). lnn.ntn was best for TD and lsn.ntn and lsn.atn were best for T.

Table 8: Results of the various weighting schemes, for English topics on CLEF 2005 test collection.. In bold are the best scores for TDN, TD, and T.

	Weighting scheme	**TDN**	**TD**	**T**
		MAP	**MAP**	**MAP**
1	lnn.ntn	**0.1366**	**0.1313**	0.1207
2	lnc.ntn	0.1362	0.1214	0.1094
3	mpc.ntn	0.1283	0.1219	0.1107
4	npc.ntn	0.1283	0.1219	0.1107
5	mpc.mtc	0.1283	0.1219	0.1107
6	mpc.mts	0.1282	0.1218	0.1108
7	mpc.nts	0.1282	0.1218	0.1108
8	npn.ntn	0.1258	0.1247	0.1118
9	lsn.ntn	0.1195	0.1233	**0.1227**
10	lsn.atn	0.0919	0.1115	**0.1227**
11	asn.ntn	0.0912	0.0923	0.1062
12	snn.ntn	0.0693	0.0592	0.0729
13	sps.ntn	0.0349	0.0377	0.0383
14	nps.ntn	0.0517	0.0416	0.0474
15	mtc.atc	0.1138	0.1151	0.1108

In all the presented experiments we used stemming when indexing the collection and the translated topics (except Section 5.3). We do not present the results here, but when we tried using an English lemmatizer (to produce base forms of inflected words) instead of a stemmer, the results were slightly worse for all settings; when using no-stemming during indexing the performance was much worse.

5.3. Comparison of Various Translations

For the cross-language task, the topics were translated into English. We used several online machine translation (MT) tools. The idea behind using multiple translations is that they might provide more variety of words and phrases, therefore improving the retrieval performance. The seven online MT systems that we used for translating from Spanish, French, and German were:

- http://www.google.com/language_tools?hl=en
- http://www.babelfish.altavista.com
- http://freetranslation.com
- http://www.wordlingo.com/en/products_services/wordlingo_translator.html
- http://www.systranet.com/systran/net
- http://www.online-translator.com/srvurl.asp?lang=en
- http://www.freetranslation.paralink.com

For the Czech language topics we were able to find only one online MT system:

- http://intertran.tranexp.com/Translate/result.shtml

The Spanish, German, and Czech topics provided by the CLEF organizers contained translations of all the fields (title, description, and narrative). For French the narrative field was not translated by the CLEF organizers, due to lack of time. An example of French query is shown in Figure 14.

<top>
<num>1159
<title>Les enfants survivants en Suède
<desc>Descriptions des mécanismes de survie des enfants nés entre 1930 et 1933 qui ont passé la guerre en camps de concentration ou cachés et qui vivent actuellement en Suède.
</top>

Figure 14: Example of French query.

We combined the outputs of the MT systems by simply concatenating all the translations. All seven translations of a title made the title of the translated query; the same was done for the description and narrative fields. An example of combined output, for the above French query shown in Figure 13, is shown in Figure 15.

Table 9 presents results for each translation produced by the seven online MT tools, from Spanish, French, and German into English on the CLEF 2005 test collection. The last column is for the combination of all translations, as explained above. All the results in the table are for lnn.ntn, TDN (except for French where only TD was available).

The translations from German and the one from Czech had many words that were not translated; they were kept unchanged into the English output of the MT tools. This would explain the lower performance for German and Czech. The MT tool number 6 for German seems to obtain better results on the test data than the combination, but this was not the case on the training data. In general, the combination of all translations performs better than the individual translations.

<top>
<num> 1159
<title> surviving children in Sweden
surviving children in Sweden
The children survivors in Sweden
surviving children in Sweden
surviving children in Sweden
The surviving children in Sweden
surviving children in Sweden
<desc> Descriptions of the mechanisms of survival of the children born between 1930 and 1933 who passed the war in concentration camps or hidden and who currently live in Sweden.
Descriptions of the mechanisms of survival of the children born between 1930 and 1933 who passed the war in concentration camps or hidden and who currently live in Sweden.
Descriptions of the survival mechanisms of the born children between 1930 and 1933 that passed the war in concentration camps or hidden and that live currently in Sweden.
Descriptions of the mechanisms of survival of the children born between 1930 and 1933 who passed the war in concentration camps or hidden and who currently live in Sweden.
Descriptions of the mechanisms of survival of the children born between 1930 and 1933 who passed the war in concentration camps or hidden and who currently live in Sweden.
Descriptions of the mechanisms of survival of the children been born between 1930 and 1933 which crossed war in concentration camps or hidden and that live in Sweden nowadays.
Descriptions of the mechanisms of survival of the children born between 1930 and 1933 who passed the war in concentration camps or hidden and who currently live in Sweden.
<narr>
</top>

Figure 15: An example of combined output, for the French query.

Table 9: Results on the output of each Machine Translation system for Spanish, French, German, and Czech on CLEF 2005 test collection.

Measure	Translation							
	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Spanish
MAP	0.111	0.115	0.115	0.116	0.115	0.118	0.115	0.1163
	Fr1	Fr2	Fr3	Fr4	Fr5	Fr6	Fr7	French
MAP	0.124	0.125	0.122	0.126	0.125	0.127	0.125	0.1285
	Gr1	Gr2	Gr3	Gr4	Gr5	Gr6	Gr7	German
MAP	0.094	0.093	0.098	0.093	0.093	0.099	0.093	0.0981
	Czech							
MAP	0.076							

Table 10 presents results for the combined translation produced by the seven online MT tools on CLEF 2006 collection, for comparison with monolingual English experiments (the first line in the table), for the topic field combinations. All the results in the table are from SMART using the lnn.ntn weighting scheme.

The retrieval results for French translations were very close to the monolingual English results, especially on the training data. On the test data, the results were much worse when using only the titles of the topics, probably because the translations of the short titles were less precise. For translations from the other languages, the retrieval results deteriorate rapidly in comparison to the monolingual results. We believe that the quality of the French-English translations produced by online MT tools was very good, while the quality was lower for Spanish, German and Czech, successively.

Table 10: Results (MAP scores) of the cross-language experiments for CLEF 2006 collection, where the indexed fields are ASRTEXT2004A, and AUTOKEYWORD2004A1, A2 using SMART (lnn.ntn).

Language	Training			Test		
	TDN	TD	T	TDN	TD	T
English	0.0954	0.0906	0.0873	0.0766	0.0725	0.0759
French	0.0950	0.0904	0.0814	0.0637	0.0566	0.0483
Spanish	0.0773	0.0702	0.0656	0.0619	0.0589	0.0488
German	0.0653	0.0622	0.0611	0.0674	0.0605	0.0618
Czech	0.0585	0.0506	0.0421	0.0400	0.0309	0.0385

5.4. Results on Phonetic Transcriptions

Part of our experiments aimed to explore the use of phoneme sequences as matching units instead of words. The word-based transcripts and queries were translated to phonemes sequences using NIST's text-to-phone tool[18], they were transformed into sequences of 4-grams, and indexed. Queries were pre-processed in a similar fashion before execution. . The intuition behind using 4-gram phone sequences is that they might have a higher noise tolerance than words, and word boundaries become less important. Figure 16 shows the output after applying the text-to-phone and splitting the sequence to 4-grams.

In Table 11 we present results for the experiment where the text of the collection and the queries

[18] www.nist.gov/speech/tools/

were transcribed into phonetic form and split into 4-grams that we used for indexing (without stemming).

We wanted to test the hypothesis that the phonetic form could help compensate for the speech recognition errors made when the collection was produced. When the fields TD were indexed, the results are better than when only T is indexed. When combining phonetic and text forms (by simply indexing both phonetic n-grams and text), the result improved compared to using only the phonetic forms, but the MAP scores are lower than the results on the text form for documents and queries. Therefore, we can conclude that on our task, the phonetic transcriptions did not help with the retrieval.

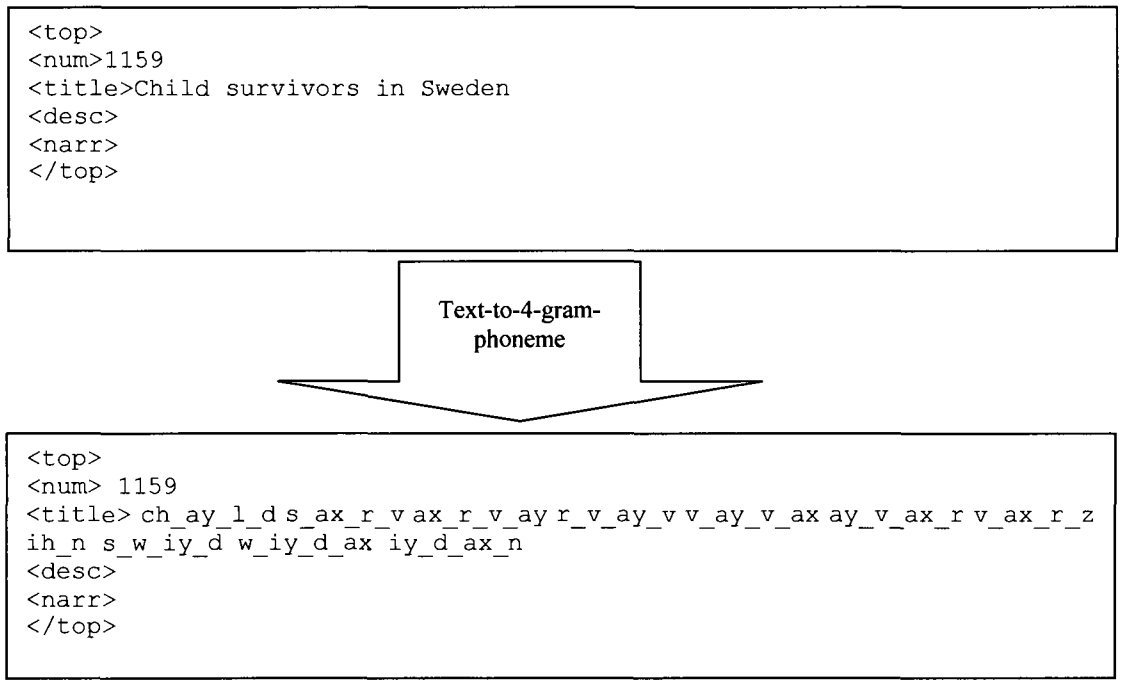

Figure 16: The output of text to 4-garam phoneme on a query title.

Table 11.Results on phonetic n-grams, and combination text plus phonetic transcripts for topics in English, and the translations from Spanish, French, German, and Czech. All the runs in this table use lnn.ntn on CLEF 2005 test collection.

Language	MAP	Fields	Description
English	0.0986	T	Phonetic
English	0.1019	TD	Phonetic
English	0.0981	T	Phonetic+Text
English	0.1066	TD	Phonetic+Text
Spanish	0.0898	T	Phonetic
Spanish	0.0972	TD	Phonetic
Spanish	0.0948	T	Phonetic+Text
Spanish	0.1009	TD	Phonetic+Text
French	0.0931	T	Phonetic
French	0.1052	TD	Phonetic
French	0.0929	T	Phonetic+Text
French	0.1072	TD	Phonetic+Text
German	0.0744	T	Phonetic
German	0.0782	TD	Phonetic
German	0.0746	T	Phonetic+Text
German	0.0789	TD	Phonetic+Text
Czech	0.0479	T	Phonetic
Czech	0.0583	TD	Phonetic
Czech	0.0510	T	Phonetic+Text
Czech	0.0614	TD	Phonetic+Text

5.5. Manual Summaries and Keywords

An interesting feature of the MALACH test collection is that ASR transcriptions are accompanied by automatically and manually-derived meta-data fields. The manual summaries consist in a few sentence that summarize a speech segment (usually three sentences), written by humans who were lis-

tening to the sound track. The manual keywords were also selected by humans, from a domain-specific thesaurus of words and phrases . It is very important to see how these manual meta-data fields affect the retrieval. We conducted different experiments by combining the ASR fields and manual meta-data fields.

Table 12 explores the effect using the manual fields: manual keywords and 3-line summaries (full manual transcripts were not available). The retrieval performance increased significantly for all topic languages (more than double). The MAP score increased from 13.66% to 32.56% when using the manual meta-data, for English TDN, for the CLEF 2005 test collection.

Table 12: Results of indexing all the fields of the CLEF 2005 test collection: the manual keywords and summaries, in addition to the ASR transcripts. Again we report results of lnn.ntn scheme because they are the best.

Language	**MAP**	**Fields**
English	0.3256	TDN
English	0.2989	TD
English	0.2754	T
Spanish	0.2548	TDN
French	0.2608	TD
German	0.2275	TDN
Czech	0.1667	TDN

Table 13 presents the results when only the manual keywords and the manual summaries were used on CLEF 2006 collection. The retrieval performance improved a lot, for topics in all the languages. The MAP score jumped from 0.0654 to 0.2902 for English test data, TDN, with the In(exp)C2 weighting model in Terrier. The results of cross-language experiments on the manual data show that the retrieval results for combined translation for French and Spanish language were very close to the monolingual English results on training data and test data. For all the experiments on manual summaries and keywords, Terrier's results are better than SMART's.

Table 13: Results of indexing the manual keywords and summaries on CLEF 2006, using SMART with weighting scheme lnn.ntn, and Terrier with (In(exp)C2).

Language and System	Training			Test		
	TDN	TD	T	TDN	TD	T
English SMART	0.3097	0.2829	0.2564	0.2654	0.2344	0.2258
English Terrier	**0.3242**	**0.3227**	**0.2944**	**0.2902**	**0.2710**	**0.2489**
French SMART	0.2920	0.2731	0.2465	0.1861	0.1582	0.1495
French Terrier	0.3043	0.3066	0.2896	0.1977	0.1909	0.1651
Spanish SMART	0.2502	0.2324	0.2108	0.2204	0.1779	0.1513
Spanish Terrier	0.2899	0.2711	0.2834	0.2444	0.2165	0.1740
German SMART	0.2232	0.2182	0.1831	0.2059	0.1811	0.1868
German Terrier	0.2356	0.2317	0.2055	0.2294	0.2116	0.2179
Czech SMART	0.1766	0.1687	0.1416	0.1275	0.1014	0.1177
Czech Terrier	0.1822	0.1765	0.1480	0.1411	0.1092	0.1201

5.6. Comparison of Systems and Query Expansion Methods

The word error rate in the MALACH test collection is about 40%, which mean a high chance that some of query terms will be missing from the ASR transcripts. So, if a relevant document does not contain the terms that are in the query, then that document will not be retrieved. This leads us to investigate query expansion methods, in order to reduce this query/document mismatch by expanding the query using words or phrases with a similar meaning or some other statistical relation to the set of relevant documents.

5.6.1 Collocations-based Query Expansion

To investigate the effects of query expansion on the retrieval, we have developed a new technique based on extracting collocations from the collection. A comparison with other techniques was conducted.

We have used the collocations-based query expansion mechanism when using SMART for retrieval. We extracted related words for each word in the topics using the Ngram Statistics Package (NSP) [155]. The query expansion procedure (we call it SMARTnsp) is defined as follows:

1) Concatenate all the ASR transcripts form MALACH test collection.
2) Remove all stop words.

3) Conflate each word in the collection to their stem by using Porter stemmer [107].
4) Extract the top 6,412 pairs of related words (collocations) based on log likelihood ratios (those with the highest collocation scores in the corpus of ASR transcripts), using a window size of 10 words. We chose log-likelihood scores because they are known to work well even when the text corpus is small.
5) For each word in the topics, add the related words from this list of pairs.

5.6.2 Thesaurus-based Query Expansion

We have a query expansion method based on a manually-built thesaurus from the Shoah Visual History Foundation, provided with the MALACH collection. Figure 16 shows two entries from the thesaurus; each entry contains six types of fields:

- name: contains a unique numeric code for each entry.
- label: a phrase or word which represents the entry.
- alt-label: contains the alternative phrase or the synonym for the entry.
- usage: contains the usage or the definition of the entry.
- There are two more relations in the thesaurus: is-a and of-type, which contain the numeric code of the entry involved in the relation.

Our method adds items and their alternatives (synonyms) from the thesaurus, based on the similarity between the thesaurus terms and the title field for each topic. More specifically, according to the following procedure:

1) Compute the similarity using SMART, where the title of each topic represents the query and the thesaurus terms as documents, using the weighting scheme lnn.ntn.
2) Add the top two thesaurus terms to the topic.
3) For each thesaurus term, add all the alternative terms to the topic.
4) Weight the original terms, thesaurus terms, and the alternative terms by the weights of 5, 2, 1 respectively, in order to give more weight to the closest terms.
5) Compute the similarity using SMART between the new weighted topics and transcripts, using the weighting scheme lnn.ntn.

For example, for topic 3,005, the title is "Death marches", and the most similar terms from the

thesaurus are "death marches" and "deaths during forced marches" As shown in Figure 17, the alternative terms for theses terms are "death march" and "Todesmärsche".

```
<keyword>
    <name>9125</name>
    <alt-label>death march</alt-label>
    <alt-label> Todesmärsche</alt-label>
    <broader-term>15445</broader-term>
    <label>death marches</label>
    <of-type>5289</of-type>
    <usage>Forced marches of prisoners over long distances, under heavy guard and
    extremely harsh conditions. (The term was probably coined by concentration
    camp prisoners.)</usage>
</keyword>
<keyword>
    <name>15460</name>
    <broader-term>15445</broader-term>
    <label>deaths during forced marches</label>
    <of-type>4109</of-type>
    <usage>The daily experience of individuals with death during forced marches
    that was not the result of executions, punishments, arbitrary killings or
    suicides.</usage>
</keyword>
```

Figure 17: The top two entries from the thesaurus that are similar to the topic title "Death marches".

5.6.3 Bose-Einstein Model for Query Expansion

Another query expansion method that we implemented extracts the most informative terms from the top-returned documents as the expanded query terms. In this expansion process, 12 terms from the returned documents (the top 15 documents) were added to the topic, based on Bose-Einstein 1 model (Bo1) [90]. See Section 2.5.1 for more details on this model.

We have put a restriction on the new terms: their document frequency must be less than the maximum document frequency in the title of the topic. The aim of this restriction is avoid more-general terms being added to the topic. Any term that satisfies this restriction will be a part of the new topic. We have also up weighted the title terms five times higher than the other terms in the topic. We run the new queries against the ASR transcripts using SMART with weighting scheme lnn.ntn.

5.6.4 Kullback-Leibler (KL) Model for Query Expansion

We have also used a query expansion mechanism together with the Terrier information retrieval system [64]. This method follows the idea of measuring the divergence from randomness, where IR is seen as a probabilistic process [62, 90]. We experimented with the $I(n_e)C2$ weighting model for retrieval and with the Kullback-Leibler (KL) model for query expansion [90]. See Section 2.5.1 for more details on these models.

5.6.5 Query Expansion Experiments Results

Table 14 presents results for the best weighting schemes: for SMART we chose lnn.ntn and for Terrier we chose the In(exp)C2 weighting model, because they achieved the best results on the training data. We present results with and without relevance feedback. According to Table 14, we note that:

- Kullback-Leibler helps to improve the retrieval results in Terrier for TDN, TD, and T for the training data; the improvement was significant for TD and T, but not for TDN. For the test data there is a small non significant improvement.
- NSP relevance feedback with SMART does not help to improve the retrieval for the training data, but it helps a little for the test data.

Table 14: Results (MAP scores) for Terrier and SMART, with or without relevance feedback, for English topics from CLEF2006 collection. In bold are the best scores for TDN, TD, and T.

System	Training			Test		
	TDN	TD	T	TDN	TD	T
SMART(lnn.ntn)	0.0954	0.0906	0.0873	0.0756	0.0711	0.0759
SMARTnsp	0.0923	0.0901	0.0870	0.0768	0.0754	0.0769
SMART+thesaurus	0.0965	0.0941	0.0901	0.0759	0.0730	0.0779
SMART+Bo1	0.0955	0.0954	0.0912	0.0776	0.0811	0.0784
SMART+thesaurus+Bo1	**0.0971**	**0.0969**	**0.0933**	**0.0812**	**0.0799**	**0.0809**
Terrier	0.0913	0.0834	0.0760	0.0651	0.0560	0.0656
TerrierKL	0.0915	0.0952	0.0906	0.0654	0.0565	0.0685

- Thesaurus-based query expansion and Bose-Einstein model help to improve the retrieval on the training and test data but not significantly.
- The combination of Thesaurus-based query expansion and Bose-Einstein model help to

improve the retrieval on the training and test data but not significantly.

- SMART results are better than Terrier results for the test data, but not for the training data.

5.7. The Effects of ASR Transcripts, Automatic Keywords, or the Combination on the Retrieval

The MALACH collection documents contain four fields that represents the output of three speech recognition system with different word error rates and named entity error rate, as shown by Table 15. The ASRTEXT2006B contains the content identical to the ASRTEXT2006A field when available and content identical to the ASRTEXT2004A field otherwise.

Table 15: Word error rates and named entity error rate for each ASR fields in MALACH collection.

	ASR field	**word error rate**	**named entity error rate**
1	ASRTEXT 2003A	40%	66%
2	ASRTEXT2004A	38%	32%
3	ASRTEXT2006A	25%	not known
4	ASRTEXT2006B	25%-38%	not known

Given these ASR fields, we are interested to investigate different hypothesis:

- In order to find the best ASR transcripts to use for indexing the segments, we compared the retrieval results when using the ASR transcripts from the years 2003, 2004, and 2006 or combinations.
- We would like to investigate the effect of a word error rate of 25-40% and named entity error rate 32-66% on retrieval performance by comparing the retrieval results from the speech recognition systems (ASR fields) vs. different error rates.
- We also wanted to investigate if we combine the ASR transcripts from different recognizer would improve the retrieval. Given that each speech recognizer system developed independently, they use different training data, vocabulary, and language models. It is possible that different recognition errors were generated, where some words may have been correctly recognized by one system and wrongly by the other. In this case merging the speech recognizer transcripts the correct recognitions of one system could compen-

sate for the wrong ones of the other system. Moreover, the weighting schemed depend on the term frequency; if a word has been correctly recognized by both systems, then it will have a larger frequency of occurrences and this will increase its weight in the context of the document. On the other hand, a word that has been wrongly recognized by one of the speech recognition systems will have a small frequency of occurrence (unless it has been consistently recognized wrongly, a case that we suppose it does not happen frequently) and therefore it will get a lower weight in the context of the document.

- The MALACH collection contains two automatic keywords fields (AUTOKEYWORD2004A1 and AUTOKEYWORD2004A2); these fields contain a set of thesaurus keywords that were assigned automatically using a k-Nearest Neighbor (kNN) classifier using only words from the ASRTEXT2004A field of the segment. We are interested to see if adding the automatic keywords helps to improve the retrieval results.

5.7.1 The Experimental Results

The results of the experiments using Terrier and SMART are shown in Table 16 and Table 17, respectively.

We note from the experimental results that:

- Using Terrier, the best field is ASRTEXT2006B which contains 7377 transcripts produced by the ASR system on 2006 and 727 transcripts produced by the ASR system in 2004, this improvement over using only the ASRTEXT2004A field is very . On the other hand, the best ASR field using SMART is ASRTEXT2004A.
- Any combination between two ASRTEXT fields does not help.
- Using Terrier and adding the automatic keywords to ASRTEXT2004A improved the retrieval for the training data but not for the test data. For SMART it helps for both the training and the test data.
- In general, adding the automatic keywords helps. Adding them to ASRTEXT2003A or ASRTEXT2006B improved the retrieval results for the training and test data.
- For the required submission run English TD, the maximum MAP score was obtained by the combination of ASRTEXT 2004A and 2006A plus automatic keywords using Terrier (0.0952) or SMART (0.0932) on the training data; on the test data the combination of

ASRTEXT 2004A and automatic keywords using SMART obtained the highest value, 0.0725.

- The name entity error rate has more effect on the retrieval than word error rate.

Table 16: Results (MAP scores) for Terrier, with various ASR transcript combinations. In bold are the best scores for TDN, TD, and T.

	Terrier						
Segment fields	**Training**			**Test**			
	TDN	**TD**	**T**	**TDN**	**TD**	**T**	**Average**
ASRTEXT 2003A	0.0733	0.0658	0.0684	0.0560	0.0473	0.0526	0.0606
ASRTEXT 2004A	0.0794	0.0742	0.0722	**0.0670**	0.0569	0.0604	0.0684
ASRTEXT 2006A	0.0799	0.0731	0.0741	0.0656	0.0575	0.0576	0.0680
ASRTEXT 2006B	0.0840	0.0770	0.0776	0.0665	**0.0576**	0.0591	0.0703
ASRTEXT 2003A+2004A	0.0759	0.0722	0.0705	0.0596	0.0472	0.0542	0.0633
ASRTEXT 2004A+2006A	0.0811	0.0743	0.0730	0.0638	0.0492	0.0559	0.0662
ASRTEXT 2004A+2006B	0.0804	0.0735	0.0732	0.0628	0.0494	0.0558	0.0659
ASRTEXT 2003A+ AUTOKEYWORD2004A1,A2	0.0873	0.0859	0.0789	0.0657	0.0570	0.0671	0.0737
ASRTEXT 2004A+ AUTOKEYWORD2004A1, A2	0.0915	0.0952	0.0906	0.0654	0.0565	0.0685	0.0780
ASRTEXT 2006B+ AUTOKEYWORD2004A1,A2	**0.0926**	0.0932	0.0909	0.0717	0.0608	0.0661	0.0792
ASRTEXT 2004A+2006A+ AUTOKEYWORD2004A1, A2	0.0915	**0.0952**	**0.0925**	0.0654	0.0565	**0.0715**	0.0788
ASRTEXT 2004A+2006B+ AUTOKEYWORD2004A1,A2	0.0899	0.0909	0.0890	0.0640	0.0556	0.0692	0.0764

Table 17: Results (MAP scores) for Terrier, with various ASR transcript combinations. In bold are the best scores for TDN, TD, and T.

	SMART						
Segment fields	**Training**			**Test**			
	TDN	**TD**	**T**	**TDN**	**TD**	**T**	**Aver-age**
ASRTEXT 2003A	0.0625	0.0586	0.0585	0.0508	0.0418	0.0457	0.0530
ASRTEXT 2004A	0.0701	0.0657	0.0637	0.0614	0.0546	0.0540	0.0616
ASRTEXT 2006A	0.0537	0.0594	0.0608	0.0455	0.0434	0.0491	0.0520
ASRTEXT 2006B	0.0582	0.0635	0.0642	0.0484	0.0459	0.0505	0.0551
ASRTEXT 2003A+2004A	0.0685	0.0646	0.0636	0.0533	0.0442	0.0503	0.0574
ASRTEXT 2004A+2006A	0.0686	0.0699	0.0696	0.0543	0.0490	0.0555	0.0612
ASRTEXT 2004A+2006B	0.0686	0.0713	0.0702	0.0542	0.0494	0.0553	0.0615
ASRTEXT 2003A + AUTOKEYWORD2004A1,A2	0.0923	0.0847	0.0839	0.0674	0.0616	0.0690	0.0765
ASRTEXT 2004A+ AUTOKEYWORD2004A1,A2	**0.0954**	0.0906	0.0873	**0.0766**	**0.0725**	**0.0759**	0.0831
ASRTEXT 2006B+ AUTOKEYWORD2004A1,A2	0.0869	0.0892	0.0895	0.0650	0.0659	0.0734	0.0783
ASRTEXT 2004A+ 2006A + AUTOKEYWORD2004A1,A2	0.0903	**0.0932**	0.0915	0.0654	0.0654	0.0777	0.0806
ASRTEXT 2004A +2006B + AUTOKEYWORD2004A1,A2	0.0895	0.0931	**0.0919**	0.0652	0.0655	0.0742	0.0799

5.8. Conclusion

In this chapter, we have explored our first attempt to investigate the retrieval from transcribed spontaneous speech.

Transcribed spontaneous speech text is differed than any regular written text collection in two ways: the first one related to the style of spoken language and the second one is related to the high error rate caused by the speech recognition system which produced the transcripts. Our first goal was to investigate many retrieval strategies to determine which one is suitable for this type of collection; therefore we have investigated different retrieval strategies (indexing schemes) from SMART and Terrier. The experiments results showed that the state-of-art retrieval strategies, such as the variation of Okapi formula performed worse than a classical weighting scheme from SMART (lnn.ntn). We believe that the reason behind this observation is that the important concepts are heavily repeated in spontaneous speech transcripts, so that the more a concept appears in the speech, the more important the concept is. In the Okapi scheme or in Terrier's indexing schemes, the frequency is normalized according to the length of the document and other statistics, which decreases the importance of these words. Therefore it was very important to select the appropriate retrieval strategy for this type of collection.

For cross-language IR, multiple query translations were combined in the hope that multiple translations provide more variety of words and phrases. We have combined the results of seven machine translation tools. The experiments showed that the combined translation is better than any individual translation involved in the combination.

The retrieval from phonetic transcriptions of the segments was investigated to test the hypothesis that the phonetic form could help compensate for the speech recognition errors made when the collection was produced. Each of the segments and the queries were translated into phoneme sequences, where 4-gram sequences were considered. The experiments results showed that phonetics transcription hurt the retrieval.

To address the mismatch problem due to speech recognition errors, we have proposed two relevance feedback methods: the first one is based on generating and adding collocations, and the second one is based on the thesaurus. We also tried to combine these methods with other state-of-the-art

methods like Kullback-Leibler and Bose-Einstein. Our experiments showed that combining the thesaurus-based method with Bose-Einstein method helps to improve the retrieval, but for any individual method the improvement was not significant.

The Mallach collection contains different transcripts from different speech recognition systems, with a varied error rate, and different types of meta-data like automatic keywords, manual summaries and manual keywords. The experiments results showed that combining the speech transcripts and the automatic keywords helps to improve the retrieval. Combining different ASR transcripts from different recognition systems during the indexing does not help to improve the retrieval. Perhaps the two speech recognition systems from IBM were similar and they misrecognized words in the same way.

In chapter 9, we will study how we could combine the results of different document representations and how the meta-data could improve the retrieval using data fusion techniques.

Chapter 6. Cluster-based Model Fusion

6.1. Introduction

Users tend to express their queries in various ways: sometimes they use more general terms, sometimes more specific terms, or a combination of both. Information retrieval systems need to be able to accommodate this variety of user needs; there is also variation among the collections (if it is a special collection like the one we use or a general collection the news collection). Some retrieval models or weighting schemes perform better when the queries are general, others perform better when the queries are more specific, and others when a combination is available. We are looking for a system that will perform well in all these cases. Figure 18 shows the variation between the weighting schemes performance according to topics, on the 2006 training data.

There are two solutions to this problem. One solution is to fuse the retrieval results of many available weighting schemes with a reasonable weight for each scheme chosen on the training data. We will investigate this approach in Section 6.2, where the model-fusion technique fuses the results of 15 weighting schemes. This system outperformed the other systems tested on the MALACH collection. This system has a drawback with regard to the running time, because it is takes a long time to run 15 weighting schemes for each query, and then to fuse the results.

In this chapter, we propose a second solution as well, that selects a smaller number of weighting schemes according to the query type, and then it fuses the results from those weighting schemes. The experiments that we will present show that having not more than 7 weighing schemes for each query type works better or as well as the fusion of the 15 weighting schemes.

The remainder of this chapter is organized as follows: Section 6.2 is pointing out the most important work in model fusion. Section 6.3 describes how we cluster the topics according to tf-idf feature. Section 6.4 describes the second fusion model that we developed. Section 6.5 presents our experimental results. Finally, Section 6.6 presents conclusions and future work.

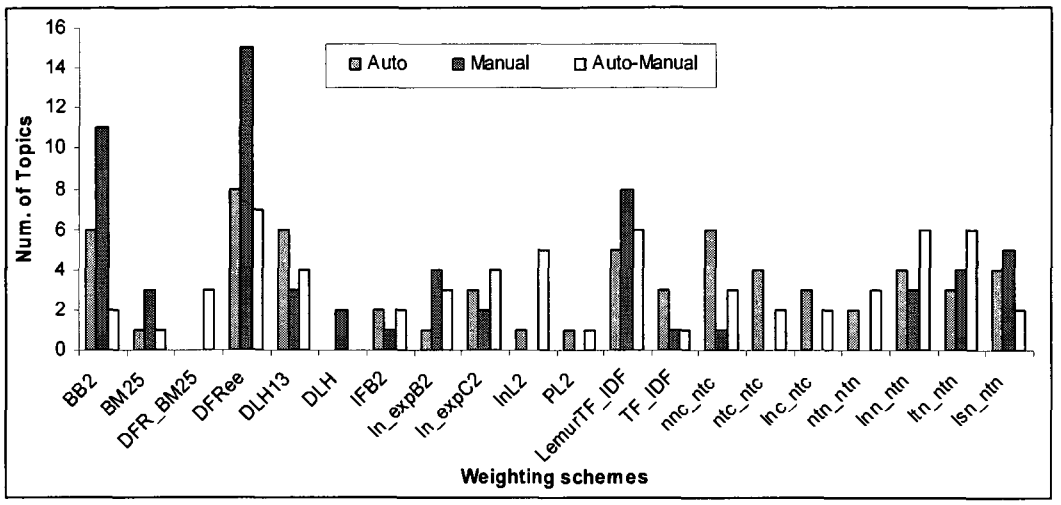

Figure 18: Variation between the weighting schemes performance according to topics, on the 2006 training data.

6.2. MAP_Recall weight for *WCombSUM* Fusion

Previous research has explored the idea of combining the results of different retrieval strategies; the motivation is that each technique will retrieve different sets of relevant documents; therefore combining the results could produce a better result than any of the individual techniques.

The classical way to fuse the results of different retrieval strategies is using *CombSUM* which sums all the scores for each document from all the strategies, as described in Section 2.5.3. If a training data is available, we should be able to specify certain belief about the quality of each retrieval strategy involved in the fusion, so we could assign predefined weight for each retrieval strategy according to the training data, and then apply these weights on the test data, this fusion method called *WCombSum* as described in section 2.5.3. In this section we are proposing two new methods for generating these predefined weights based on the training data, and then apply these weights in *WCombSUM* formula. These weight generated based on the MAP and recall values. Basically, our fusion technique is the summation of normalized weighted similarity scores of 15 different IR schemes (strategies), where the weights are generated based on the MAP and Recall from the training data, as shown in the following formulas:

$$W_1 CombSUM = \sum_{i \in IR\ schems} [W_r^4(i) + W_{MAP}^3(i)] * NormalizedScore_i \qquad (6.1)$$

$$W_2 CombSUM = \sum_{i \in IR\ schems} W_r^4(i) * W_{MAP}^3(i) * NormalizedScore_i \qquad (6.2)$$

where $W_r(i)$ and $W_{MAP}(i)$ are experimentally determined weights based on the recall (the number of relevant documents retrieved) and precision (MAP score) values for each IR scheme computed on the training data. For example, suppose that two retrieval runs r1 and r2 give 0.3 and 0.2 (respectively) as MAP scores on training data; we normalize these scores by dividing them by the maximum MAP value: then $W_{MAP}(r1)$ is 1 and $W_{MAP}(r2)$ is 0.66 (then we compute the power 3 of these weights, so that one weight stays 1 and the other one decreases; we chose power 3 for MAP score and power 4 for recall, because the MAP is more important than the recall). We hope that when we multiply the similarity values with the weights and take the summation over all the runs, the performance of the combined run will improve. $NormalizedScore_i$ is the normalized similarity for each IR scheme. We did the normalization by dividing the similarity by the maximum similarity in the run. The normalization is necessary because different weighting schemes will generate different range of similarity values, so a normalization method should applied to each run. Our method is differed than the work done by Fox and Shaw in 1994 [4], and Lee in 1995 [6]; they combined the results by taking the summation of the similarity scores without giving any weight to each run. In our work we weight each run according to the precision and recall on the training data.

6.2.1 Experiments Using MAP_Recall weight for *WCombSUM* Fusion

We applied the data fusion methods (*W_1CombSUM and W_2CombSUM*) described in above to 14 runs produced by SMART and one run produced by Terrier. Performance results for each single run and the results of the fused runs are presented in Table 18, in which % change is given with respect to the run providing better effectiveness in each combination on the training data. The Manual English column represents the results when only the manual keywords and the manual summaries were used for indexing the documents using English topics, the Auto-English column represents the results when automatic transcripts are indexed from the documents, for English topics. For cross-languages experiments the results are represented in the columns Auto-French, and Auto-Spanish, when using the combined translations produced by the seven online MT tools, from French and Spanish into English. Since the result of combined translation for each language was better than when using individual translations from each MT tool on the training data [8], we used only the combined translations in our experiments.

Data fusion helps to improve the performance (MAP score) on the test data. The best improvement using data fusion (*W_1CombSUM*) was on the French cross-language experiments with 21.7%,

which is statistically significant while on monolingual the improvement was only 6.5% which is not significant. We computed these improvements relative to the results of the best single-model run, as measured on the training data. This supports our claim that data fusion improves the recall by bringing some new documents that were not retrieved by all the runs as shown in Table19. On the training data, the *W_2CombSUM* method gives better results than *W_1CombSUM* for all cases except on Manual English, but on the test data Fusion1 is better than *W_2CombSUM*. In general, the data fusion seems to help, because the performance on the test data in not always good for weighting schemes that obtain good results on the training data, but combining models allows the best-performing weighting schemes to be taken into consideration.

Experimental results on training and test data showed that our fusions methods (*W_1CombSUM and W_2CombSUM*) significantly outperform the methods proposed by Fox and Shaw in [12], see Section 2.5.3 for more details about their methods. On the other hand, CombMNZ performs better than their other methods as shown in Figure 19 and Figure 20.

Table 18: Results (MAP scores) for 15 weighting schemes using Smart and Terrier (the In(exp)C2 model), and the results for the two Fusions Methods(W1CombSUM and W2CombSUM). In italic are the best scores for the 15 single runs on the training data and the corresponding results on the test data. Underlined are the results of the runs that we submitted to CLEF-CLSR 2007.

Weighting scheme	**Manual English**		**Auto-English**		**Auto-French**		**Auto-Spanish**	
	Train.	**Test**	**Train.**	**Test**	**Train.**	**Test**	**Train.**	**Test**
nnc.ntc	0.2546	0.2293	0.0888	0.0819	0.0792	0.055	0.0593	0.0614
ntc.ntc	0.2592	0.2332	0.0892	0.0794	0.0841	0.0519	0.0663	0.0545
lnc.ntc	0.2710	0.2363	0.0898	0.0791	0.0858	0.0576	0.0652	0.0604
ntc.nnc	0.2344	0.2172	0.0858	0.0769	0.0745	0.0466	0.0585	0.062
anc.ntc	0.2759	0.2343	0.0723	0.0623	0.0664	0.0376	0.0518	0.0398
ltc.ntc	0.2639	0.2273	0.0794	0.0623	0.0754	0.0449	0.0596	0.0428
atc.ntc	0.2606	0.2184	0.0592	0.0477	0.0525	0.0287	0.0437	0.0304
nnn.ntn	0.2476	0.2228	0.0900	0.0852	0.0799	0.0503	0.0599	0.061
ntn.ntn	0.2738	0.2369	0.0933	0.0795	0.0843	0.0507	0.0691	0.0578
lnn.ntn	0.2858	0.2450	**0.0969**	**0.0799**	0.0905	0.0566	0.0701	0.0589
ntn.nnn	0.2476	0.2228	0.0900	0.0852	0.0799	0.0503	0.0599	0.0610
ann.ntn	0.2903	0.2441	0.0750	0.0670	0.0743	0.0380	0.0570	0.0383
ltn.ntn	0.2870	0.2435	0.0799	0.0655	0.0871	0.0522	0.0701	0.0501
atn.ntn	0.2843	0.2364	0.0620	0.0546	0.0722	0.0347	0.0586	0.0355
In(exp)C2	**0.3177**	**0.2737**	0.0885	0.0744	**0.0908**	**0.0487**	0.0747	0.0614
W_1CombSUM	0.3208	0.2761	0.0969	0.0855	0.0912	0.0622	0.0731	0.0682
% change	1.0%	0.9%	0.0%	6.5%	0.4%	21.7%	-2.2%	10.0%
W_2CombSUM	0.3182	0.2741	0.0975	0.0842	0.0942	0.0602	0.0752	0.0619
% change	0.2%	0.1%	0.6%	5.1%	3.6%	19.1%	0.7%	0.8%

Table 19: Results (number of relevant documents retrieved) for 15 weighting schemes using Terrier and SMART, and the results for the Fusions Methods. In bold are the best scores for the 15 single runs on training data and the corresponding test data; underlined are the runs that we submitted to CLEF-CLSR 2007.

Weighting scheme	Manual English		Auto-English		Auto- French		Auto- Spanish	
	Train.	Test	Train.	Test	Train.	Test	Train.	Test
nnc. ntc	2371	1827	1726	1306	1687	1122	1562	1178
ntc.ntc	2402	1857	1675	1278	1589	1074	1466	1155
lnc.ntc	2402	1840	1649	1301	1628	1111	1532	1196
ntc.nnc	2354	1810	1709	1287	1662	1121	1564	1182
anc.ntc	2405	1858	1567	1192	1482	1036	1360	1074
ltc.ntc	2401	1864	1571	1211	1455	1046	1384	1097
atc.ntc	2387	1858	1435	1081	1361	945	1255	1011
nnn.ntn	2370	1823	1740	1321	**1748**	**1158**	1643	1190
ntn.ntn	2432	1863	1709	1314	1627	1093	1502	1174
lnn.ntn	2414	1846	1681	1325	1652	1130	1546	1194
ntn.nnn	2370	1823	**1740**	**1321**	1748	1158	**1643**	**1190**
ann.ntn	2427	1859	1577	1198	1473	1027	1365	1060
ltn.ntn	2433	1876	1582	1215	1478	1070	1408	1134
atn.ntn	2442	1859	1455	1101	1390	975	1297	1037
ln(exp)C2	**2638**	**1823**	1624	1286	1676	1061	1631	1172
W_1CombSUM	2645	1832	1745	1334	1759	1147	1645	1219
% change	0.3%	0.5 %	0.3%	1.0%	0.6%	-1.0%	0.1%	2.4%
W_2CombSUM	2647	1823	1727	1337	1736	1098	1631	1172
% change	0.3%	0.0%	0.8%	1.2%	-0.7%	-5.5%	-0.7%	-1.5%

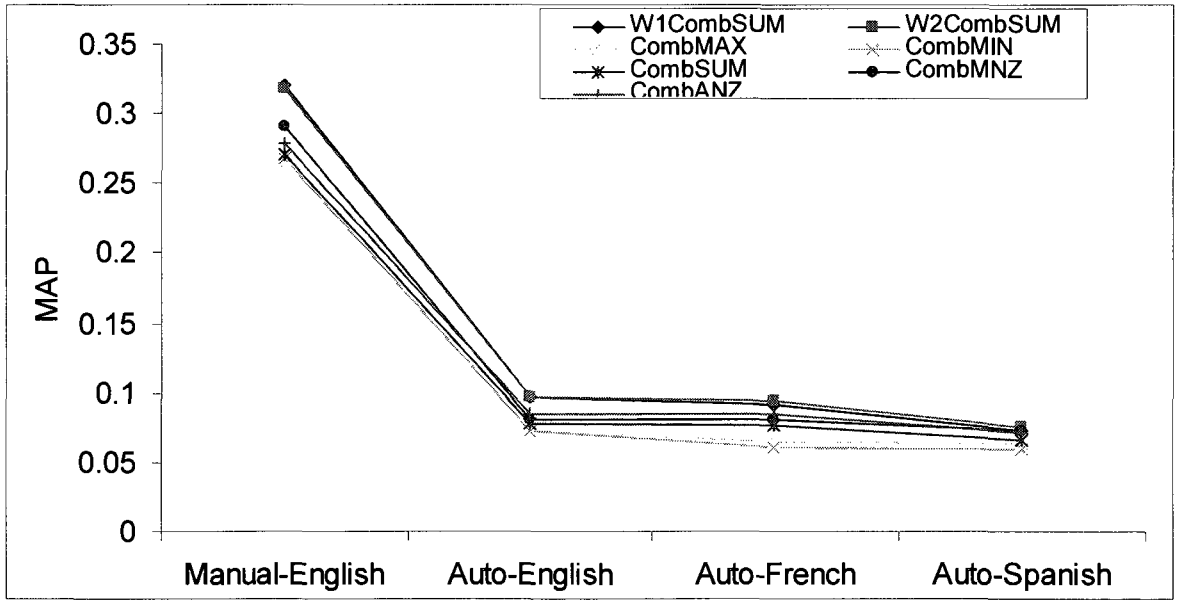

Figure 19: Comparing the results of *W_1CombSUM* and *W_2CombSUM* to the methods proposed by Fox and Shaw on the training data of CLEF-CLSR 2007.

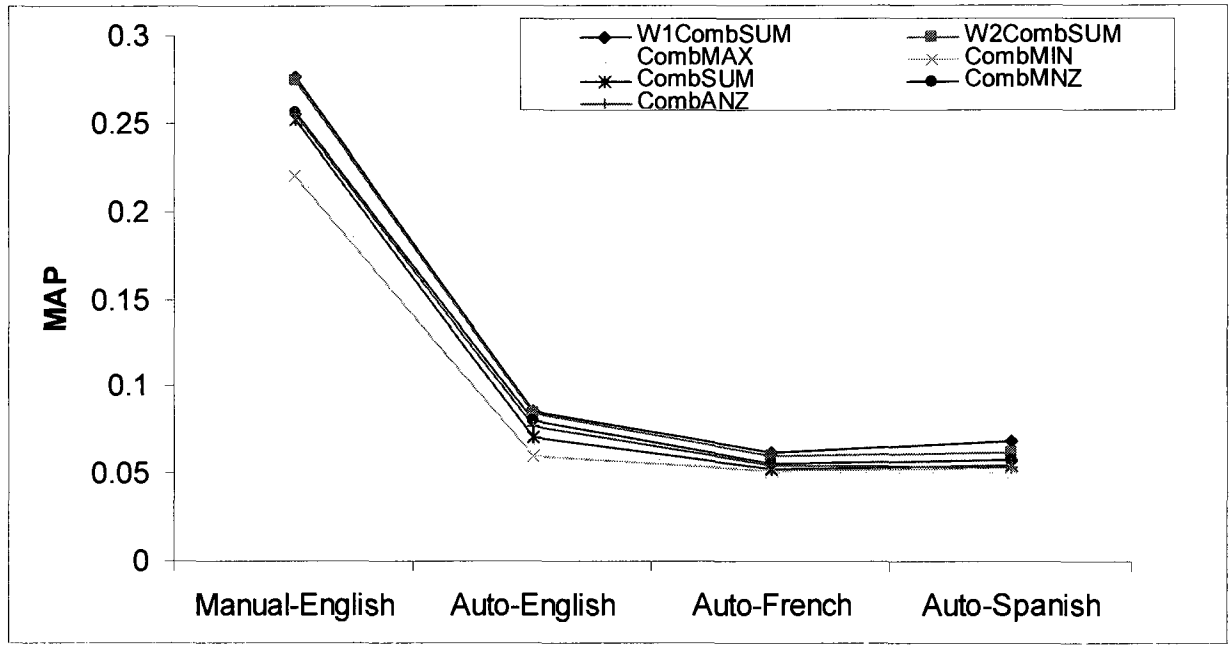

Figure 20: Comparing the results of $W_1CombSUM$ and $W_2CombSUM$ to the methods proposed by Fox and Shaw on the test data of CLEF-CLSR 2007.

6.3. Features selection, Clustering, and Fusion

In this section, we will explore the idea of combining the results of different retrieval strategies, according to characteristics of the user's query. We propose a novel data fusion technique for combining the results of different IR models. We choose a feature based on TF-IDF (term frequency-inverse document frequency) that allows us to cluster the training queries/topics from the collection. Then we select the best weighting schemes for each cluster as a combination of the best scheme for each of the topics from the cluster. Later on we use this feature to classify the test topics into the appropriate clusters and run the corresponding combination of weighting schemes.

6.3.1 Features and Clustering

One important issue is what query features to consider when clustering the training topics. Once we decided on what are the clusters into which we arrange the training topics, we will select

the best weighting schemes for each cluster and fuse them. When the system runs on a test topic, it will determine into which cluster to categorize the new topic, and it will run the combination of weighting schemes that was previously determined for that cluster.

Looking at the experimental results for the weighting schemes mentioned in section 6.3 on the training topics, we noticed a variation between the weighting schemes performance according to topics. Each of the weighting schemes performs better than other weighting scheme on some topics; moreover, the best weighting scheme in terms of Mean Average Precision (MAP score) on the training queries (DFree) is not the same as the best weighting scheme on the test topics (nnc.ntc). These two observations guide us to propose a new model fusion method that performs better than every single weighting scheme across all the topics. We hypothesize that there are clusters of topics so that each cluster prefers some specific weighting schemes. To prove this we have to find features that will allow us to cluster the training topics.

Our proposed feature is based on the TF-IDF values of the terms in the queries. This feature has four parts that weight each term in the query[19]:

- The term frequency in the collection, that is in all documents, (tf_c), which can be calculated by formula (6.3), where TF is the term frequency in the document collection (how many times the term occurs in the collection), DF is the document frequency (how many documents the term occurs in), and $Max_{tfc}(q)$ is the maximum tfc for any term in the query. We divide by Maxtfc(q) to normalize the values, and we multiply by the log(Maxtfc(q)) to increase the weight for terms that appeared more frequently in the document. The intuition behind this part is that the more often the term appears in the document, the more important the term is.

$$tf_c = \frac{TF/DF}{MAX_{tfc}(q)} * \log(MAX_{tfc}(q)) \qquad (6.3)$$

- The inverse document frequency (idf), which can be calculated by formula (6.4), where N is the total number of documents in the collection, and DF is the document frequency. The intuition behind this part is to include the discrimination power of each term, i.e., a term that appears in fewer documents is more discriminate.

$$idf = \log(\frac{N}{DF}) \qquad (6.4)$$

[19] We experimented with several formulas and this one was the best on the training data.

- Term frequency of the term in the query (tfq): which can be calculated by formula (6.5), where tf is the term frequency in the topic, and $MAX_{tfq}(q)$ is the maximum tf in the topics. We divide by $MAX_{tfq}(q)$ to normalize the value. The intuition behind the term frequency part is that the more often the term appears in the topic, the more important the term is.

$$tf_q = \frac{tf}{MAX_{tfq}(q)} \qquad (6.5)$$

- The length normalization part, which can be calculated by formula (6.6), represents the total number of terms in the topic. We use this part in order to get an average value for all the terms in the topic.

$$len = \frac{1}{\sum tf} \qquad (6.6)$$

Then the feature weight (FW) of the query is calculated by formula (6.7), which is the summation for each part of the feature for each term in the topic.

$$FW = \sum_{for\ each\ term\ in\ topic} tf_c * idf * tf_q * len \qquad (6.7)$$

After calculating the feature weight FW for each training topic, it is time to cluster the topics. We use one of the most popular clustering techniques, the K-Mean method. For this method we have to decide how many clusters we are looking for. Therefore we tried different numbers of clusters, and we chose 15 because the output of the clustering method gave us clusters with a maximum size of 7, which is a reasonable number of weighting schemes to fuse, assuming that each cluster prefers 7 weighing schemes at most.

6.3.2 *WRCombMNZ* Model Fusion

Our model fusion formula is a modified version of the method proposed by [108]; their method, called *CombMNZ*, sums up all the normalized scores of a document multiplied by the number of non-zero scores of the document, for more detail, see section 2.5.3 .

For each cluster of topics, as described in section 4, there are some retrieval strategies preferred by the cluster, and these retrieval strategies have different MAP scores. For that reason we adapt *combMNZ* to carry a weight for each weighting scheme in the cluster. Our cluster-based fusion model uses a fusion formula that we call *WRCombMNZ* represented by the following formual:

$$WRCombMNZ = \sum_{i \in IR\ schemes} W_{ik} * Normalizedscore_i * n \qquad (6.8)$$

where W_{ik} is a pre-calculated weight associated with each weighting scheme's results in the cluster k, n is the number of non-zero scores of the document, and the NormalizedScorei is calculated by formula 2.72 as described before.

The weight (W_{ik}) for each weighting scheme is calculated based on the MAP score for each cluster on the training data, reflecting how much its cluster prefers this weighting scheme, using the following formula:

$$W_{ik} = \begin{cases} 1 & \text{if the weighting scheme k has the max MAP} \\ & \text{for at least two topics in the cluster} \\ 1 & \text{if the weighting scheme k has the max median} \\ & \text{MAP for all the topics in the cluster k} \\ 0.1 & \text{otherwise} \end{cases} \qquad (6.9)$$

Basically, our model fusion with this particular weights allow the best weighting scheme to contribute the most, and the others to support it by two contributions: the first one is a small factor (0.1) of the normalized score of the document, and the second one helps re-rank the document proportional to the number n of the non-zero scores of the document. The intuition behind using 1 as a weight for some weighting schemes is that in some clusters there is more than one topic that prefers a particular weighting scheme; this is a strong indication that in these cluster this weighting scheme is one of the best in the cluster. Another case is when each topic in the cluster prefers a different scheme. In this case we select the weighting scheme with the maximum median MAP score among the topics in the cluster to have the weight one. The reason for selecting the median, not the mean, is because the median is less sensitive to the extreme MAP scores and a better indicator for smaller sample size, while the mean is often used with larger sample.

Our cluster-based model fusion differs from other works in the literature in that we fuse the retrieval results based on clusters of weighting schemes, and in the way we weight each weighting scheme(retrieval strategies) for each cluster.

6.3.3 Experimental Results using WRCombMNZ

The candidate retrieval strategies (weighting schemes) for our fusion system were provided by two IR systems: SMART [49, 156] and Terrier [62, 64] . We used the (nnc.ntc, ntc.ntc, lnc.ntc, ntn.ntn, lnn.ntn, ltn.ntn, lsn.ntn) weighting schemes from SMART [49, 156], and (BB2, BM25, DFR_BM25, DFRee, DLH13, DLH, IFB2, In_expB2, In_expC2, InL2, PL2, LemurTF_IDF, and TF_IDF); for more detail about these technique see chapter 2.

We applied the *K-mean* clustering method on the 63 training topics. 15 clusters were produced based on tf-idf values, as described in section 4. For each cluster, from 7 runs produced by SMART and 13 runs produced by Terrier, the best weighting scheme for each topic was selected based on its MAP score. The weight for each weighting schemes for each cluster was calculated based on the MAP score, as described in section 6.5. After that we applied the data fusion method for the best weighting schemes of the particular cluster, as described in section 6.5. Maximum 7 weighting scheme was fused for each cluster because there were maximum 7 topics in each cluster. Then each test topic was classified based on its tf-idf value into one of the 15 clusters previously-produced based on the training data and the data fusion formula for the cluster was applied.

We conducted three types of experiments, based on the fields which were indexed. In the first one, the automatic transcripts (ASRTEXT2006B), and two automatic keywords (AK1 and AK2) were used for indexing the documents; we call this experiment Auto. In the second experiment, we indexed the manual keywords and the manual summaries for each document; we named this experiment Manual. In the last experiment we indexed the automatic transcripts, the two automatic keywords fields, the manual summaries, and the manual keywords; we call this experiment Auto+Manual. The title and description fields from each topic are used as query. Table 20 shows some statistics about each experiment. One interesting observation is that the number of terms (distinct words) in the manual fields is about half of the number of terms in the automatic fields. The number of tokens (total number of words) in the manual fields is about 16% of the number of tokens in the automatic fields. The average term frequencies are 39, 125, and 125 for Manual, Auto, and Auto+Manual, respectively. This ratio is very high, about four times more in the Auto fields. We also note that combining Auto and Manual brings about 14% of the terms to the Auto+Manual list of terms, which means that there is more information in the combined fields.

Table 20. Some statistics about the number of terms and the number of tokens for the three experiments.

	Number of terms	Number of tokens
Auto	13,605	1,711,684
Manual	7,131	278,717
Auto + Manual	15,884	1,990,401

Experiments on the 63 training topics using 20 weighting schemes form SMART and Terrier showed a variation between the weighting schemes performance according to topics. Each of the weighting schemes performs better than other weighting schemes on some topics. This observation guided us to propose the new model fusion technique described in section 5. Figure 18 illustrates this observation, by showing how many topics preferred by each weighting scheme.

Our experiments showed a strong relation between the feature weight (based on tf-idf) for each topic and the performance of the topics (measured as MAP score). When the value of the feature weight increased among the clusters, the maximum summation of the MAP score increased as well. Figure 21 shows the relation between the clusters and the maximum summation of the MAP score; as we see the histogram is negatively skewed, which means the smaller values are to the left and the larger values are to the right, so the maximum summation of the MAP score is increasing, and the feature weight between clusters is increasing as well, i.e. topics in cluster 1 have lower feature weights than topics in cluster 5. This proves our claim that the proposed tf-idf feature is a very good feature to cluster the topics.

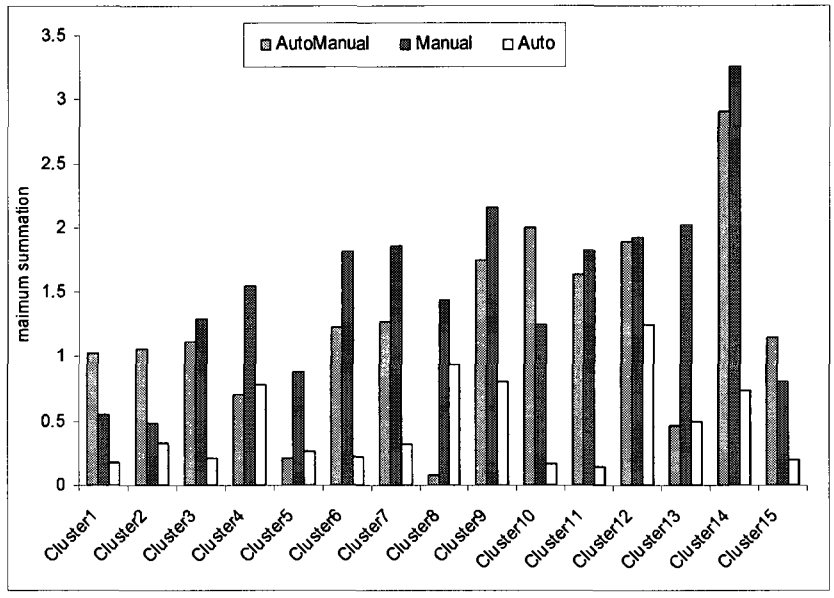

Figure 21: The relation between the clusters and the maximum summation of the MAP score.

Performance results for each single run and fused runs are presented in Table 21. The results are presented in the format MAP score, R-Precision, and number of relevant documents retrieved. In the table, % change is given with respect to the run that was best on a single model on the training data and the one on the test data.

We can conclude that cluster-based model fusion (*WRCombMNZ*) helps to improve the MAP score on the held-out test data. The improvement is statistically significant comparing to all individual weighting schemes, based on a one-tailed Wilcoxon signed rank test with ($p < 0.05$), except for nnc.ntc, ln_expB2, and nnc.ntc for Auto, Manual, and Auto+Manual, respectively, for which the improvement was only statistically significant with $p<0.1$. Moreover, the results was significantly better ($p<0.05$) comparing to the best weighting schemes on the training data (Dfree). It is very important to compare with the best system on the training data because the researchers often select the system based on the training data. The best improvement using the cluster-based model fusion was on the Auto experiments with 9%, 22% relative changes comparing to the best system on the test data and the training data, respectively. Also, there is an improvement in the number of relevant documents retrieved (Recall) and R-Precision for all the experiments (see in Table 21). This supports our claim that data fusion improves the recall by bringing some new documents that were not retrieved by all the runs. Moreover, the improvement on MAP score means that the data fusion method gives a better ranking for the documents in the list. One very important observation is that the best

weighting scheme on the training data is not the best weighting scheme on the test data. For example for the Auto experiment, DFree was the best on the training data, and nnc.ntc on the test data. In general, the data fusion helps, because the performance on the test data is not always good for weighting schemes that obtain good results on the training data, but combining models allows the best-performing weighting schemes for each cluster to be taken into consideration.

Experiments show that our cluster based model fusion performs better than the individual weighting schemes for different levels of recalls. Figures 22, 23, and 24 show Precision-Recall graphs for 11 levels of recall for the three experiments: Auto, Manual, and Auto+Manual, respectively, in order to compare our model fusion method (*WRCombMNZ*) with the best weighting scheme on the training data and on the test data.

Table 21: Results (MAP scores, R-Precision, and number of relevant documents retrieved) for 20 weighting schemes from Smart and Terrier, and the results of the fusion methods, on the test data. In bold we marked the best weighting scheme on the test data, and underlined is the best weighting scheme on the training data (though we do not show the actual results on the training data, that we used for development).

Weighting scheme	**Auto**			**Manual**			**AutoManual**		
	MAP	**R-Prec.**	**Rel.-Ret.**	**MAP**	**R-Prec.**	**Rel.-Ret.**	**MAP**	**R-Prec.**	**Rel.-Ret.**
BB2	0.0441	0.0793	972	0.2699	0.3113	1826	0.0970	0.1280	1133
BM25	0.0567	0.0952	1120	0.2490	0.2911	1824	0.1404	0.1793	1381
DFR_BM25	0.0580	0.0984	1122	0.2558	0.2993	1818	0.1408	0.1815	1407
DFRee	0.0695	0.1179	1298	0.2527	0.2840	1822	0.1586	0.2056	1697
DLH13	0.0735	0.1162	1335	0.2560	0.2968	1825	0.1606	0.2078	1720
DLH	0.0719	0.1162	1325	0.2460	0.2904	1812	0.1606	0.2039	1707
IFB2	0.0605	0.1016	1080	0.2705	0.3025	1824	0.135	0.1831	1335
In_expB2	0.0657	0.1099	1259	**0.2727**	**0.3063**	**1826**	0.1537	0.2077	1581
In_expC2	0.0700	0.1144	1288	0.2704	0.3127	1826	0.1551	0.2103	1609
InL2	0.0629	0.1020	1259	0.2575	0.3000	1826	0.1521	0.1951	1570
PL2	0.0730	0.1172	1295	0.2510	0.2887	1803	0.1575	0.1991	1658
LemurTF_IDF	0.0517	0.0894	1146	0.2269	0.2674	1814	0.1319	0.1753	1425
TF_IDF	0.0651	0.1044	1302	0.2525	0.2965	1818	0.1452	0.1938	1627
nnc_ntc	**0.0779**	**0.1210**	**1270**	0.2190	0.2525	1760	**0.161**	**0.2047**	**1698**
ntc_ntc	0.0630	0.1097	1235	0.2154	0.2691	1776	0.1525	0.1994	1623
lnc_ntc	0.0722	0.1190	1269	0.2270	0.2863	1784	0.1585	0.2111	1667
ntn_ntn	0.0649	0.1161	1250	0.2140	0.2614	1792	0.1464	0.1951	1643
lnn_ntn	0.0658	0.1169	1284	0.2346	0.2808	1789	0.1527	0.2100	1684
ltn_ntn	0.0512	0.0924	1166	0.2167	0.2601	1785	0.1297	0.1810	1511
lsn_ntn	0.0426	0.0792	1028	0.1856	0.2312	1787	0.1140	0.1517	1376
Fusion	0.0849	0.1325	1353	0.2801	0.3191	1848	0.1671	0.2261	1720
%change (test)	**9%**	**10%**	**7%**	**3%**	**4%**	**1%**	**4%**	**10%**	**1%**
%change (train)	22%	12%	4%	10%	12%	1%	5%	9%	1%

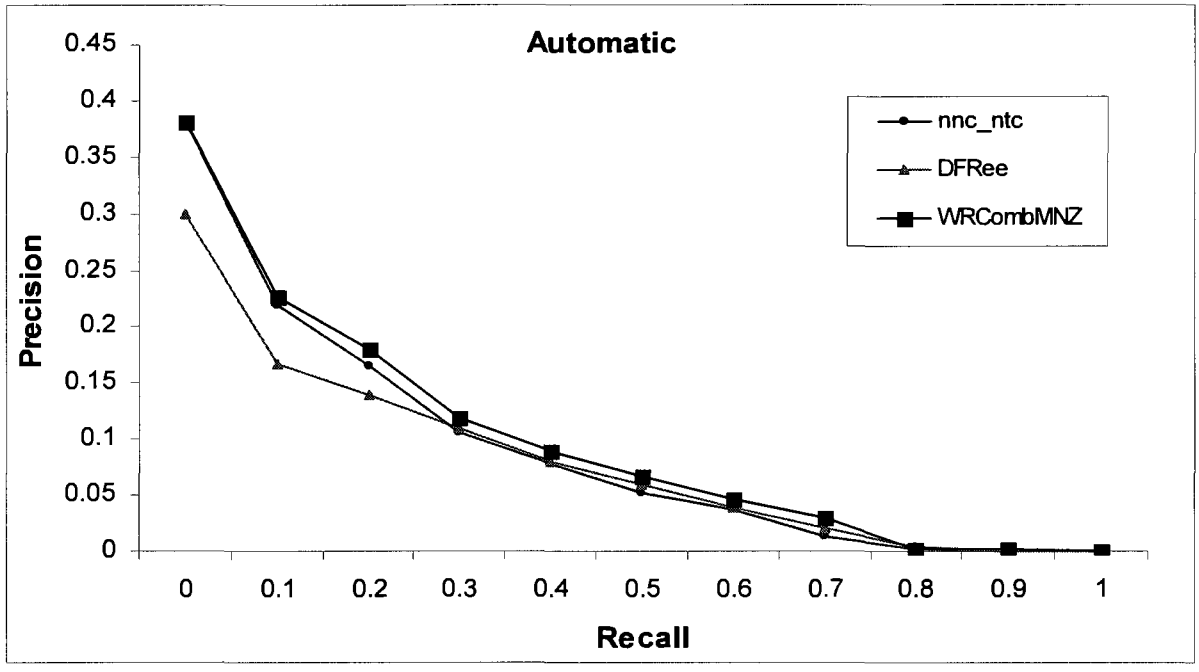

Figure 22: Precision-Recall graph for 11 levels of recall for Auto experiment.

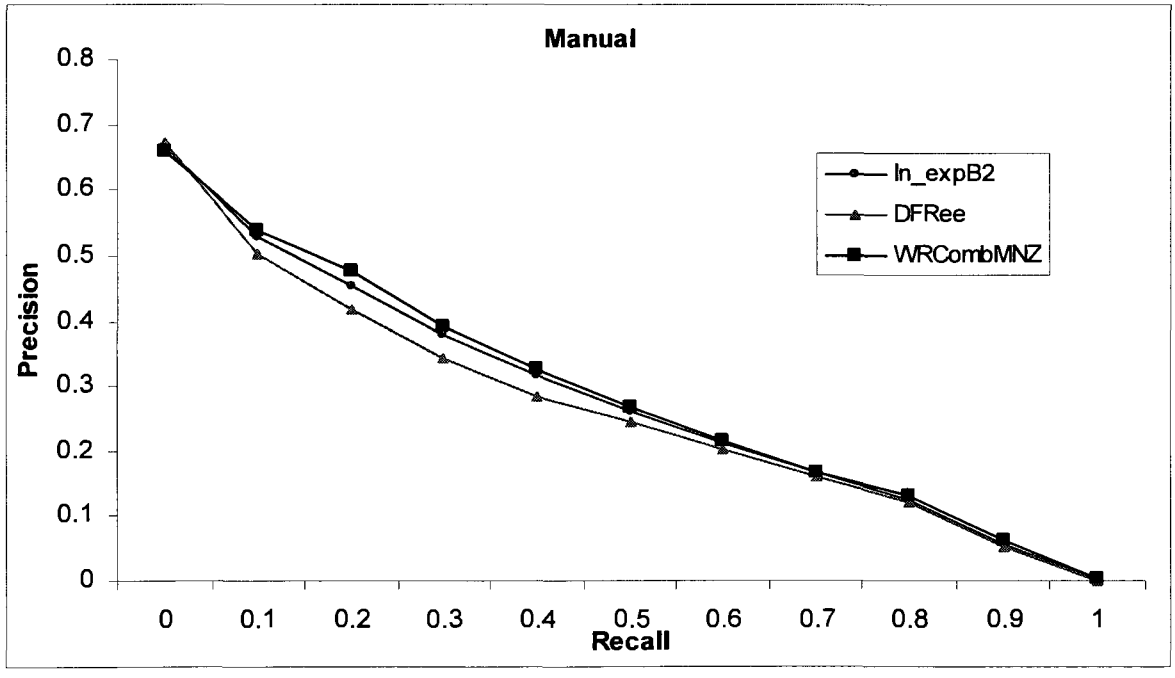

Figure 23: Precision-Recall graph for 11 levels of recall for Manual experiment.

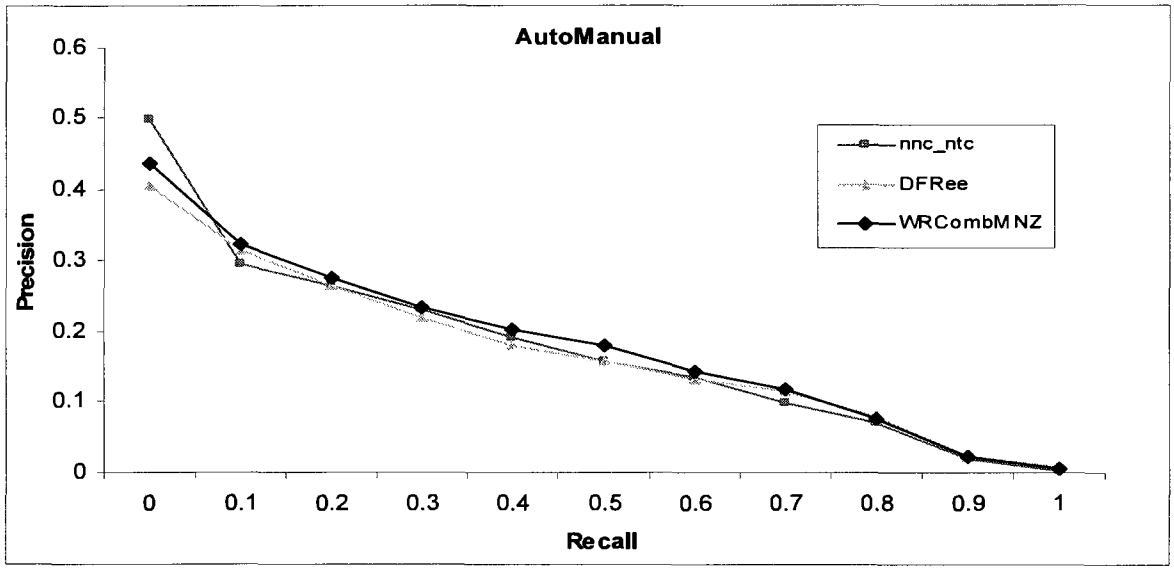

Figure 24: Precision-Recall graph for 11 levels of recall for Auto+Manual experiment.

We can compare our new method (*WRCombMNZ*) with the results of other IR systems on the same test set (using the 33 English test queries and the automatic transcripts – the required run for the CLSR task at CLEF 2007). For this setting we obtained a MAP score of 0.0849. This result was approximately the same as the best system proposed by Alzghool and Inkpen [157] (the MAP score was 0.855), see section 6.2 for more detail. It can be considered better because this system has a drawback with regard to the running time, because it is takes a long time to run 15 weighting schemes for each query, and then to fuse the results. Moreover, our cluster-based fusion technique is better than the other 4 systems that participated in the task [158], as reported in Table 22.

Table 22: Results for our system and the 5 teams that participated in the CLSR task at CLEF 2007, on the English test queries.

Submitted run	MAP score
Cluster-based fusion	0.0849
UO	0.0855
DCU	0.0787
BLLIP	0.0785
UC	0.0571
UVA	0.0444

6.4. Conclusion

In this chapter, we explored two ideas for model fusion. The first one, fusing as many as available retrieval strategies results by generating reasonable weight for each retrieval strategy based on the recall and MAP score. This idea proved to be effective, but it has drawback, it need longer time as the number of retrieval strategies increased.

The second idea that we explored in this chapter is clustering the topics in order to determine the best combination of weighting schemes for each cluster. The clusters contain words with similar levels of specificity, because they are formed according the average tf-idf of the words. We showed that the improvement achieved on the training data carries on to the test data.

One question that arises from our experiments is the following: is there any room for improvements over the results that we obtained? If yes, what are the possible ways to do this?

To answer these questions, we build an upper bound approximation for our experiments, by taking the MAP score for the best weighting scheme for each topic, then compute the average over all the topics. Results show that the upper bounds are 0.1016, 0.3082, and 0.22 for Auto, Manual, and Auto+Manual, respectively, on the test data. So, if we succeed to build a system that selects the best weighting scheme for each topic, we will get the above results. We could get more than that if our technique fuses the best weighting scheme with others that perform well for that topic. So, one way to improve our system is by clustering topics in such a way that all the topics in one cluster have the same preferred weighting scheme. This would require discovering other features for clustering.

Another way to improve our technique is to include more weighting schemes form different IR models, for example based on language modeling which has a very successful way to deal with missing terms from the query by using different smoothing techniques.

Chapter 7. Probability-based Model Fusion

7.1. Introduction

Probability theory plays a role in all studies of natural processes across all scientific disciplines. The need for a theoretical probabilistic foundation is obvious, since natural variation effects all measurements, observations and findings about different phenomena. Probability theory provides the basic techniques for statistical inference. In this chapter, we will apply the basic fundamentals rules of probability theory to derive two probabilistic-based models for fusion.

The remainder of this chapter is organized as follows: Section 7.2 is pointing out relevant related work. Section 6.3 describes the derivation of probabilistic model fusion. Section 6.4 presents our experimental results. Finally, Section 6.5 presents conclusions.

7.2. Related Work

Model fusion attempts to combine the retrieval results of different retrieval strategies[13-16, 108, 109] or retrieval results using different document representations [15]. The classical combination (CombSUM) sums up the similarity scores for each document retrieved among for all the retrieval strategy involved in the fusion [108]. The drawback of these approaches is the lack of a theoretical background. Another work done by Montague [159] relate the fusion problem to the social choice theory, where different voting algorithms such as Borda Count and Condorecet's algorithm are proposed to build model fusion systems; in these algorithms the rank of the document in the ranked list is used instead of the similarity score of the document. Also, Montague [159] attempts to solve the fusion problem by deriving a probabilistic model based on Bayesian model. The ranks are used to estimate the probability instead of the similarity scores, to compute the final score of the document the odds of relevance are computed, where the Bayes's rule is applied, finally the odds of relevance were computed according to the following formula:

$$\log O_{rel} = \sum \log \frac{P(r_i(d) \mid rel)}{P(r_i(d) \mid irr)} \qquad (7.1)$$

where $P(r_i(d) \mid rel)$ is the probability that document d would be given rank r_i by system i if it were relevant, $P(r_i(d) \mid irr)$ is the probability that d would be given rank r_i by system i if it were irrelevant.

Hull et al. [160] provide an alternative method of computing the log odd of relevance with different way to estimate the probability.

One interesting things about these methods is that they solve the problem of normalization (since the probability is between 0 and 1), and the problem un-retrieved documents score estimation.

In our experiments, we will use the classical fusion method (CombSUM) and the best system on test data as baseline methods to compare it to our new techniques.

7.3. Probability-based Model Derivation

We start the derivation of our approaches for probabilistic-based model fusion by the following example, and then we translate the problem in this example to an IR problem.

Example: Assume that we have a big box contains three small boxes (i.e. Box 1, Box 2, and Box 3), each box contains different type of balls (red, blue, yellow, black, green, white, and gray). All the balls are identical, except for their color, as shown in Figure 25. The sample space of the experiment is S={Box1{7 red, 6 blue, 5 yellow, 4 black, 3 green}, Box2{7 blue, 6 yellow, 5 red, 4 gray, 3 white}, Box3{7 gray, 6 red, 5 white, 4 green, 3 blue}}. We conduct two experiments.

In the first experiment, we draw one ball from the big box. We need to estimate the probability that the ball is red, P(red), and the probability that the ball is green, P(green).

We could compute these probabilities in two ways. One way is to compute the relative frequency of red and green balls to the total number of balls in the big box; so the P(red) and P(green) are 18/75 and 7/75, respectively. The second way is by using the law of total probability. The three small boxes form the partitions of the sample space $S = \{Box_1, Box_2, Box_3\}$, where all the events that form the partitions are mutually exclusive and $\bigcup_{i-1}^{n} Box_i = S$. Box_i is the event of selecting Box_i from the small boxes. Since the pre-conditions are satisfied, then P(red) and P(green) can be computed by:

$$P(red) = \sum_{i=1}^{3} P(red \mid Box_i) \times P(Box_i) = \frac{7}{25} \times \frac{1}{3} + \frac{5}{25} \times \frac{1}{3} + \frac{6}{25} \times \frac{1}{3} = \frac{18}{75} \quad (7.2)$$

$$P(green) = \sum_{i=1}^{3} P(green \mid Box_i) \times P(Box_i) = \frac{3}{25} \times \frac{1}{3} + \frac{0}{25} \times \frac{1}{3} + \frac{4}{25} \times \frac{1}{3} = \frac{7}{75} \quad (7.3)$$

In the second experiment, we draw three balls from the big box, one from each small box. T. What is the probability that the red ball is selected from the three boxes is $P(red_1 \cap red_2 \cap red_3)$, where red_1, red_2, and red_3 are the events of drawing a red ball from Box_1, Box_2, and Box_3 respectively. What is the probability that the green ball is selected from the three boxes $P(green_1 \cap green_2 \cap green_3)$?

We note that the event of drawing a ball from one box does not affect the drawing from the other two boxes. This means that the three events -red_1, red_2, and red_3- are statistically independent. So we can use the multiplication rule we compute

$P(red_1 \cap red_2 \cap red_3)$ and $P(green_1 \cap green_2 \cap green_3)$ by applying the following formulas:

$$P(red_1 \cap red_2 \cap red_3) = \prod_{i=1}^{3} P(red_i) = \frac{7}{25} \times \frac{5}{25} \times \frac{6}{25} = \frac{210}{15625} = 0.01344 \quad (7.4)$$

$$P(green_1 \cap green_2 \cap green_3) = \prod_{i=1}^{3} P(green_i) = \frac{3}{25} \times \frac{0}{25} \times \frac{4}{25} = \frac{0}{15625} = 0 \quad (7.5)$$

Table 23 shows the probability values for all the colors in the first and second experiment.

Table 23 : The probability values after applying the law of total probability and the law of multiplication for independent events to compute the probabilities of different colors in the boxes.

Color	$P(color)$	$P(\bigcap_{i=1}^{3} color_i)$
Red	0.24	0.01344
Blue	0.21333	0.008064
Yellow	0.14667	0
Black	0.05333	0
Green	0.09333	0
White	0.10667	0
Gray	0.14667	0

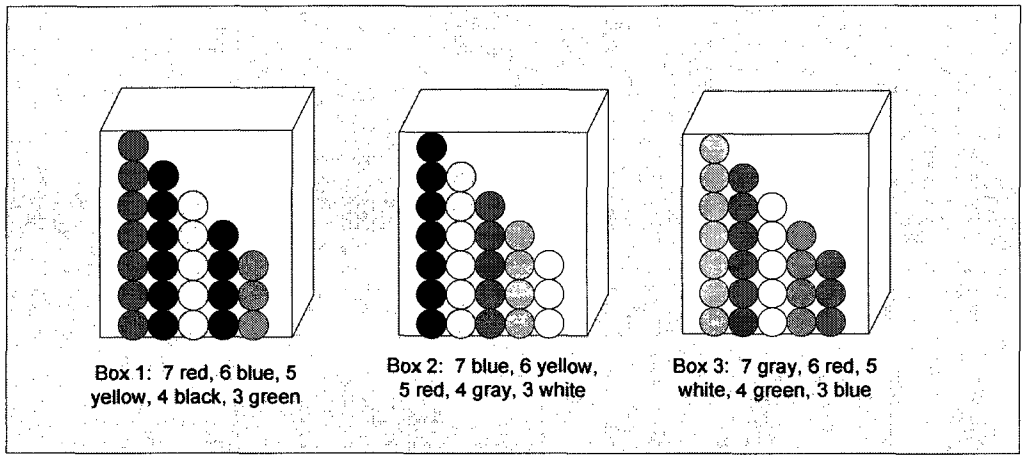

Figure 25: Example of an experiment of drawing a ball from a large box, and three balls from the smaller boxes, one from each box.

We use the previous probabilistic example to formulate the fusion or combination the results of different information retrieval models. Suppose $S = \{s_1, s_2, s_3, \cdots, s_m\}$ is a set of information retrieval systems that use different weighting schemes (different retrieval models); $D = \{d_1, d_2, d_3, \cdots, d_n\}$ is a set of documents, and $Q = \{q_1, q_2, q_3, \cdots, q_k\}$ is a set of user queries. If we run the systems on the set of documents, for one query at a time, the result will be a set of a decreasing order ranked lists of documents for each system $L = \{L_1, L_2, L_3, \cdots, L_m\}$, where $L_m = \{r_1, r_2, r_3, \cdots, r_n\}$ is the set of ranks produced by the system s_m for the query and r_n is the reversed-rank produced for document d_n. The document on the top of the raked list has the score n, meaning that this document is the most relevant document to the query. There is a restriction that each system will retrieve only the first 1000 relevant documents, which means that the ranks of the other documents will be missing from the list. We can translate the two experiments of drawing balls from the previous example to our information retrieval problem, to derive the model fusion methods. We will treat the documents as balls, systems as boxes, and frequency of particular ball in a box as the rank of the document in the ranked list.

The first fusion method is based on selecting a list from L, randomly, and then picking up a document d_j with probability relative to its reversed-rank in the list. Therefore, a document with a high reversed-rank is more probable to be selected than a lower ranked document. In other words, we are going to calculate the probability $P(d_i)$ of drawing each document from the set of ranked

lists, and then sorting the documents according to the probability $P(d_i)$ of their selection according to the random selection procedure; specifically, the fusion formula is based on the law of total probability. The set of ranked lists $S = \{L_1, L_2, L_3, \cdots, L_m\}$ produced by the set of systems forms the partitions of the sample space, where all the events that form the partitions are mutually exclusive and $\bigcup_{i-1}^{n} L_i = S$. Since the pre-conditions are satisfied, $P(d_i)$ can be computed by:

$$CombTotPROB = P(d_k) = \sum_{i=1}^{m} P(d_k \mid L_i) \times P(L_i) = \sum_{i=1}^{m} \frac{r_{ki}}{\sum_{k=1}^{n} r_{ki}} \times \frac{1}{m} \qquad (7.6)$$

where r_{ki} is the rank of document d_k from the ranked list L_i.

There are two differences between our proposed model (*CombTotPROB*) and the model proposed by Fox and Shaw in 1993 (*CombSUM*)[12]:

- Our method has theoretical background based on probabilistic theory, unlike Fox and Shaw's method.
- We have used the ranks to derive the probabilities, while Fox and Shaw used the similarity value from the systems; therefore they had to normalize the values of the similarity to be between 0 and 1, because the different systems generate different range of similarities. In our proposed method, we do not need to do the normalization, since the probability values are between 0 and 1.

In the second fusion method, we randomly and independently choose a document from each list with a selection probability within-list proportional to the reversed-rank within list. Thus the probability of selecting a document d_j simultaneously from the m lists is given by the product of probabilities of the selection within-list. In other words, the model is based on calculating the probability that the document is selected by all the systems $P(\bigcap_{i=1}^{m} d_{ki})$, where d_{ij} is the event of drawing document d_k from the rank list L_j. The events d_{k1}, d_{k2}, ..., d_{km} are statistically-independent because the event of selecting a document from one ranked list does not affect the selecting from the other ranked lists. This means that we can use the multiplication rule to compute $P(\bigcap_{i=1}^{m} d_{ki})$, and then sort the documents according to their probability. Our proposed model *CombMultPROB* is based

on the multiplication of statistically independent events and can be calculated by the following formula:

$$CombMultPROB = P(\bigcap_{i=1}^{m} d_{ki}) = \prod_{i=1}^{m} P(d_{ki}) = \prod_{j=1}^{m} \frac{r_{ki}}{\sum_{k=1}^{n} r_{ki}} \qquad (7.7)$$

The main issue with the probability estimation (maximum likelihood estimation) in our model *CombMultPROB* is that each system will retrieve the first top 1000 documents, which means that the probability estimation will be zero for the documents that are not in the ranked list. Remember from the previous example, due to this problem five out of seven colors had zero for the probability values. Intuitively, we would like the unseen balls/documents to have non-zero probabilities. One simple way to solve this problem is by adding one to all ranks in the ranked list including the missing documents. This smoothing algorithm is called Laplace smoothing or Laplace Law [161] which is given by the following formula:

$$P_{Lap}(d_{ki}) = \frac{r_{ki} + 1}{(\sum_{k=1}^{n} r_{ki}) + n} \qquad (7.8)$$

The process of adding one has the effect of giving a little bit of the probability space to unseen events. One drawback for Laplace estimate is that it gives too much of the probabilities to unseen events (overestimates). A commonly-adopted solution was proposed by Lidston's law of success [162], where we add a smaller positive value λ, instead of one:

$$P_{Lid}(d_{ki}) = \frac{r_{ki} + \lambda}{(\sum_{k=1}^{n} r_{ki}) + (n \times \lambda)} \qquad (7.9)$$

Table 24 shows the probabilities values after applying Laplace's estimation on the problem of drawing balls from the example on *CombTotPROB* and *CombMultPROB*. Note that the probability values were decreased for the seen balls and became non-zero for the unseen balls. We note also that $CombMultPROB_{Lap}$ helps to differentiate between two balls that are gray and yellow, where on the other hand *CombTotPROB* gave the two balls the same probability values. This gives us an indication that we could improve retrieval performance (MAP scores) by using $CombMultPROB_{Lap}$.

Table 24: The probability values after applying the law of total probability ($CombTotPROB$) and the law of multiplication for independent event ($CombMultPROB$) using the smoothing formula (Laplace) to compute the probabilities of different colors in boxes.

	$CombTotPROB$	$CombTotPROB_{Lap}$	$CombMultPROB$	$CombMultPROB_{Lap}$
Red	0.24	0.21875	0.01344	0.010254
	0.21333333	0.197917	0.008064	0.006836
Yellow	0.14666667	0.145833	0	0.001282
Gray	0.14666667	0.145833	0	0.001221
White	0.10666667	0.114583	0	0.000732
Green	0.09333333	0.104167	0	0.00061
Black	0.05333333	0.072917	0	0.000153

In the literature, according to our knowledge, there are two attempts to formulate the combination based on probability. One was proposed by Hull et al. [160] and the other one by Montague [159]. Both techniques were based on Bayesian analysis by estimating the logarithmic odds of relevance as probability. In our view, there is a major shortcoming in the estimation of the probability that violates the probability theory, because the probability summation of all the events will not add up to 1.

7.4. Probability-based Fusion Model Experiments

The candidate retrieval strategies (weighting schemes) for our fusion system were provided by two IR systems: SMART [49, 156] and Terrier [62, 64] . We used the (nnc.ntc, ntc.ntc, lnc.ntc, ntn.ntn, lnn.ntn, ltn.ntn, lsn.ntn) weighting schemes from SMART [49, 156], and (BB2, BM25, DFR_BM25, DFRee, DLH13, DLH, IFB2, In_expB2, In_expC2, InL2, PL2, LemurTF_IDF, and TF_IDF); for more detail about these technique see chapter 2.

We conducted three types of experiments on MALACH collection -see chapter 3 for more detail about the collection - based on the fields which were indexed. In the first one, the automatic transcripts (ASRTEXT2006B), and two automatic keywords (AK1 and AK2) were used for indexing the documents; we call this experiment Auto. In the second experiment, we indexed the manual keywords and the manual summaries for each document; we named this experiment Manual. In the last experiment we indexed the automatic transcripts, the two automatic keywords fields, the manual summaries, and the manual keywords, we call this experiment Auto+Manual. The title and description fields from each topic are used as query.

We have applied the probability-based models (CombTotPROB and CombMultPROB) to fuse the results of 20 retrieval strategies (weighting schemes) from SMART and Terrier. As baselines, we have chosen the best pre-fusion runs in the fusion process and the fusion results using the classical fusion technique CombSUM. Experiments results (see in Table 25) showed that:

- The performance of CombTotPROB and CombMultPROB are the same.
- The probability-based fusion helps to improve the retrieval comparing to the best retrieval run on the Auto+Manual segment representation, but the improvement was not significant.
- The probability-based fusions fail to improve the retrieval comparing to the best run on Auto and Manual segment representations. The decrease in the retrieval effectiveness was significant.
- The probability-based fusions perform better than CombSUM for Auto and Auto+Manual representations, but not for Manual. The increase or the decrease is not significant.

Table 25: Results (MAP scores, R-Precision, and number of relevant documents retrieved) for 20 weighting schemes from Smart and Terrier, and the results of the probability based methods, on the test data. In bold we marked the best weighting scheme on the test data.

Weighting scheme	Auto		Manual		Auto+Manual	
	MAP	Rel.-Ret.	MAP	Rel.-Ret.	MAP	Rel.-Ret.
BB2	0.0441	972	0.2699	1826	0.0970	1133
BM25	0.0567	1120	0.2490	1824	0.1404	1381
DFR_BM25	0.0580	1122	0.2558	1818	0.1408	1407
DFRee	0.0695	1298	0.2527	1822	0.1586	1697
DLH13	0.0735	1335	0.2560	1825	0.1606	1720
DLH	0.0719	1325	0.2460	1812	0.1606	1707
IFB2	0.0605	1080	0.2705	1824	0.135	1335
In_expB2	0.0657	1259	**0.2727**	**1826**	0.1537	1581
In_expC2	0.0700	1288	0.2704	1826	0.1551	1609
InL2	0.0629	1259	0.2575	1826	0.1521	1570
PL2	0.0730	1295	0.2510	1803	0.1575	1658
LemurTF_IDF	0.0517	1146	0.2269	1814	0.1319	1425
TF_IDF	0.0651	1302	0.2525	1818	0.1452	1627
nnc_ntc	**0.0779**	**1270**	0.2190	1760	**0.161**	**1698**
ntc_ntc	0.0630	1235	0.2154	1776	0.1525	1623
lnc_ntc	0.0722	1269	0.2270	1784	0.1585	1667
ntn_ntn	0.0649	1250	0.2140	1792	0.1464	1643
lnn_ntn	0.0658	1284	0.2346	1789	0.1527	1684
ltn_ntn	0.0512	1166	0.2167	1785	0.1297	1511
lsn_ntn	0.0426	1028	0.1856	1787	0.1140	1376
CombSUM	0.0691	1314	0.2637	1844	0.1635	1680
CombTotPROB	0.0693	1318	0.2599	1837	0.1649	1681
CombMultPROB	0.0692	1317	0.2599	1837	0.1649	1681

As a conclusion, probability-based fusions do not help to improve the retrieval for this collection, but they perform as the same level as the classical fusion technique CombSUM. Unlike CombSUM, the probability-based fusions have a theoretical justification and they do not need to normalize the similarity scores because the normalization is included in the model definition.

7.5. Conclusion

In this chapter, we explore the idea of combining the results of different retrieval strategies in a new way, based on the probability theory, where basic probability rules are applied to combine the results of different IR models. Unlike the classical technique, the probability-based technique does not need to normalize the score, since the probability values are between 0 and 1. The probability based techniques perform the same as the classical fusion technique.

Chapter 8. Heuristic Algorithm for Weight Selection

8.1. Introduction

A large number of IR systems and retrieval strategies have been proposed and implemented in the last 30 years. There is a tremendous need to benefit from these strategies. One way to benefit from them is to combine their results by a data fusion technique.

In chapter 6, we have proposed a model fusion technique to fuse the retrieval results of many available weighting schemes with a reasonable weight for each scheme, chosen on the training data. This technique fuses the results of 15 weighting schemes. This system outperformed the other systems when tested on the MALACH collection [11]. The system has a drawback with regard to the running time, because it takes a long time to run 15 weighting schemes for each query, and then to fuse the results.

Fusion as many as possible retrieval strategy has another critical situation which caused false alarm and wrong answer; when many of the retrieval strategies retrieve similar results or approximately the same, so it is very important to include different retrieval strategies that produce different rank lists. for example, if we have three retrieval strategies S1, S2, S3; and it happen that S2 and S3 produce the same rank list, and the rank produced by S1 is better than the one produced by S2 or S3, the results for fusing the output of the three system will be biased to the rank list produced by S2 or S3. This is not better than the best system involved in the fusion process S1. To avoid this problem, we are looking for technique that excludes similar retrieval strategies.

In this chapter we will propose a fusion technique, that selects a smaller number of retrieval strategies using a heuristic algorithm, and then it fuses the results from those strategies. The experiments that we will present show that having not more than 6 retrieval strategies works better than or as well as the fusion of the 15 weighting schemes.

The remainder of this chapter is organized as follows: Section 8.2 is pointing out relevant related work. Section 8.3 describes the heuristic algorithm for weight selection. Section 8.4 presents our experimental results. Finally, Section 8.5 presents conclusions.

8.2. Related Work

The classical way to fuse the results of different retrieval strategies is using *CombSUM* which sums all the scores for each document from all the strategies, as described in Section 2.5.3. If a training data is available, we should be able to specify certain belief about the quality of each retrieval strategy involved in the fusion, so we could assign predefined weight for each retrieval strategy according to the training data, and then apply these weights on the test data, this fusion method called *WCombSum* as described in section 2.5.3.

In the literature, there are different ways to assign a weight (W_{ik}) for each retrieval strategy:

- Manually-weighted scheme [16, 118], where the researchers try different weight values for each retrieval strategy and select the best combination. We believe this technique is an unsystematic way to derive the weights.
- MAP-based weighted scheme[31, 159, 163], where the MAP score for each retrieval strategy on training data is considered as a weight for that strategy. This technique is simple and proves to be effective for some cases when there is no performance variation between the retrieval strategies on different data.
- MAP-Recall weighted scheme [10], where the MAP and the recall score are combined to derive the weight for each retrieval strategy so that the best weighting scheme contribute the most, and the others only support it.

In section 8.4, we will propose a novel technique to train the weight for each retrieval strategy.

In our experiments, we will use CombSUM and WCombSUM as baseline method, to compare it to our new technique. As a base case, we will consider the MAP scores as the weights in the training phase for WCombSUM.

8.3. Heuristic Algorithm for Weight Selection

In section 8.2, we have reviewed different ways to derive a weight using training data for each retrieval strategy. The most unsystematic way to derive the weight was to try different weight combinations on training data, then to apply the particular combination on the test data.

Suppose we have 20 retrieval strategies, the weight for each retrieval strategy is an integer number varied from zero to four; by a simple calculation, we have to try 9.5 *1013 combinations to select the best combination for the training data. Can you imagine how long it will take to try all the

combination to find the best combination on training data? If we assume the fusion time for each combination is one second, then the total time to try all the combinations is about 30,660,826 years. This solution is for sure impossible. Therefore, we are looking for a weight selection algorithm that reduces the time for training and improves the retrieval.

Our proposed heuristic algorithm attempts to fuse the output for each pair of retrieval strategies; if the MAP score of the fused output is better than each individual output and better than the average MAP of all the 20 retrieval strategies, then this combination will survive for the next level of fusion; we will continue this procedure until a level where no combination survives for the next level. So, the best combination in the previous level will be the one to apply on test data; the weight of each retrieval strategy is the frequency of the retrieval strategy in the best combination. For example, the best combination of one of our runs is:

((((S01 S12)(S05 S12))((S01 S13)(S04 S12)))(((S01 S12)(S05 S12))((S01 S19)(S05 S13))))

According to this combination, the weights system S01, S04, S05, S12, S13, and S19 are 4, 1, 3, 5, 2, and 1 respectively. The weight for any system that does not appear in the combination is zero. As we notice the algorithm stops after the 4t-h level, and only five retrieval strategies are involved in the combination. Figure 26 shows the detailed procedure for our proposed heuristic algorithm for weight selection. In our experiments, when we use the heuristic algorithm to derive the weights, we will use the prefix "WH" before the method name; i.e., WHCombSUM.

```
Suppose S = {s1, s2, s3, ..., sm} is a set of IR systems' output (ranklist).
sij←F(si, sj) is a fusion function that fuses a pair of systems' output
(si and sj) and produces a new output sij.
1.  S = {s1, s2, s3, ..., sm}
2.  L ← 0
3.  Level_L ← S
4.  Evaluate and find the MAP score for each system in Level_L
5.  Avg ← find the average MAP of all systems in Level_L
6.  For each pair si and sj in Level_L
7.  sij ← F(si, sj)
8.  evaluate sij and find the MAP for sij
9.  if the MAP of sij is greater than to the MAP of si, sj, and avg
10.         add sij to level_L+1
11. if level_L+1 is empty
12.    return the best system combination in Level_L
13. else
14.    L ← L+1
15.    go to 4
```

Figure 26. Heuristic algorithm for weight selection.

8.4. Heuristic Algorithm for Weight Selection Experiments

The candidate retrieval strategies (weighting schemes) for our fusion system were provided by two IR systems: SMART [49, 156] and Terrier [62, 64] . We used the (nnc.ntc, ntc.ntc, lnc.ntc, ntn.ntn, lnn.ntn, ltn.ntn, lsn.ntn) weighting schemes from SMART [49, 156], and (BB2, BM25, DFR_BM25, DFRee, DLH13, DLH, IFB2, In_expB2, In_expC2, InL2, PL2, LemurTF_IDF, and TF_IDF); for more detail about these techniques see chapter 2.

We conducted three types of experiments on MALACH collection -see chapter 3 for more detail about the collection - based on the fields which were indexed. In the first one, the automatic transcripts (ASRTEXT2006B), and two automatic keywords (AK1 and AK2) were used for indexing the documents; we call this experiment Auto. In the second experiment, we indexed the manual keywords and the manual summaries for each document; we named this experiment Manual. In the last experiment we indexed the automatic transcripts, the two automatic keywords fields, the manual summaries, and the manual keywords; we call this experiment Auto+Manual. The title and description fields from each topic are used as query.

Experiments on the 63 training topics using 20 weighting schemes form SMART and Terrier showed a variation between the weighting schemes performance according to topics. Each of the

weighting schemes performs better than other weighting schemes on some topics. This observation guided us to propose the new heuristic algorithm for weight selection described in section 8.4. Figure 27 illustrates this observation, by showing how many topics preferred each weighting scheme.

Performance results for each single run and fused runs results are presented in Table 26. The results are presented in the format MAP score and number of relevant documents retrieved.

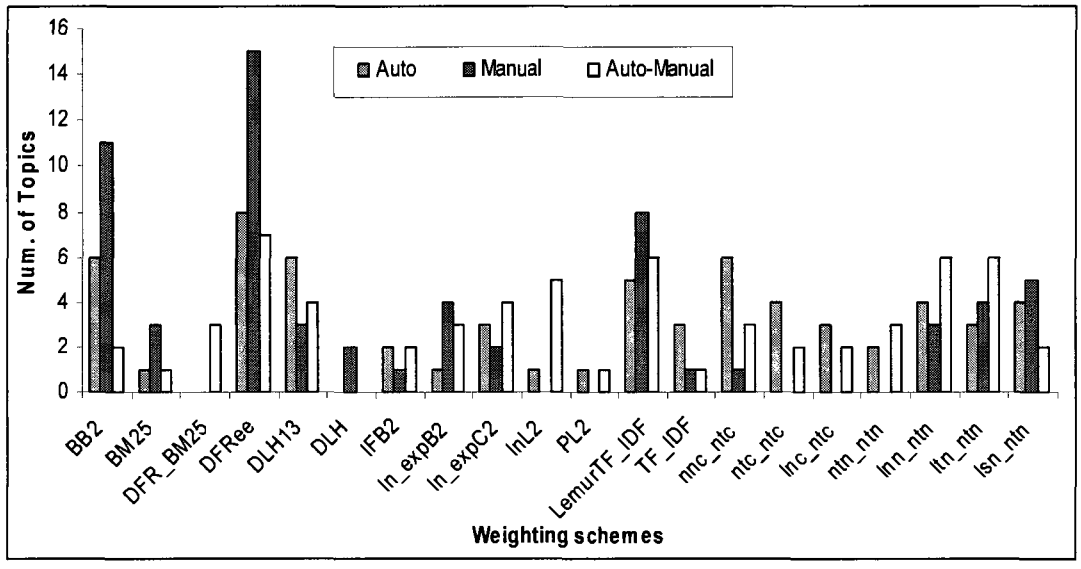

Figure 27 : Variation between the weighting schemes performance according to topics, on the 2006 training data.

As shown in Table 26 and Figure 28, the classical fusion technique (CombSUM), and even the weighted versions like WCombSUM where the weight for each retrieval strategy is the MAP on the training data failed to improve the retrieval comparing to the best retrieval strategy involved in the fusion process. Nonetheless, our heuristic algorithm for weight selection helps to improve the MAP score on the held-out test data. The improvement is statistically significant comparing to all individual weighting schemes, based on a one-tailed Wilcoxon signed rank test with ($p < 0.05$), except for the Manual experiments, for which the improvement was only statistically significant with $p<0.1$. Moreover, the results was significantly better ($p<0.05$) comparing to (CombSUM or WCombSUM). The best improvement using the heuristic-based fusion was on the Auto experiments with 10%, and 24% relative difference changes comparing to the best system on the test data and CombSUM, respectively (see in Figure 30). Also, there is an improvement in the number of relevant

documents retrieved (Recall) for all the experiments (see in Table 26). This supports our claim that data fusion improves the recall by bringing some new documents that were not retrieved by all the runs. Moreover, the improvement in MAP score means that the data fusion method gives a better ranking for the documents in the list. One very important observation is that the best weighting scheme on the training data is not the best weighting scheme on the test data. For example for the Auto experiment, DFree was the best on the training data, and nnc.ntc on the test data. In general, the data fusion helps, because the performance on the test data is not always good for weighting schemes that obtain good results on the training data, but combining models allows the best-performing weighting schemes to be taken into consideration.

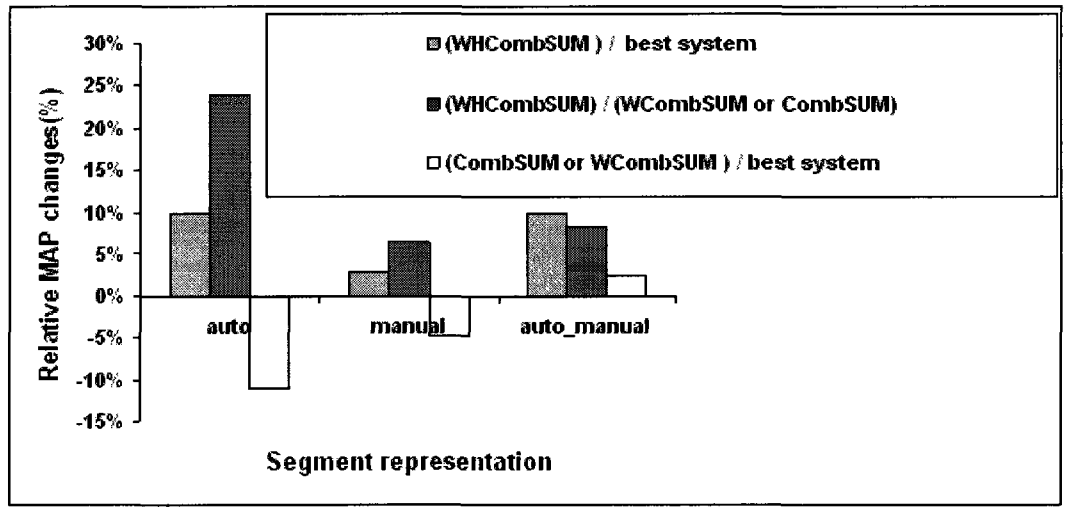

Figure 28: Comparing the heuristic-based fusion (WHCombSUM) and the classical ones (WCombSUM or CombSUM) by showing the relative MAP changes between (WHCombSUM) and the best pre-fusion run, between (WHCombSUM) and (WCombSUM or CombSUM that give almost the same results therefore the ratio will the same at two significant digits), and between (WCombSUM or CombSUM) and the best pre-fusion run. The fusion is applied to three segment representations - Auto, Manual, and Auto+Manual - to fuse 20 retrieval strategies from SMART and Terrier.

Table 26: Results (MAP scores, and number of relevant documents retrieved) for 20 weighting schemes from Smart and Terrier, and the results of the CombSUM, WCombSUM, and WHCombSUM, on the test data. In bold we marked the best weighting scheme on the test data.

Weighting scheme	Auto		Manual		Auto+Manual	
	MAP	Rel.-Ret.	MAP	Rel.-Ret.	MAP	Rel.-Ret.
BB2	0.0441	972	0.2699	1826	0.0970	1133
BM25	0.0567	1120	0.2490	1824	0.1404	1381
DFR_BM25	0.0580	1122	0.2558	1818	0.1408	1407
DFRee	0.0695	1298	0.2527	1822	0.1586	1697
DLH13	0.0735	1335	0.2560	1825	0.1606	1720
DLH	0.0719	1325	0.2460	1812	0.1606	1707
IFB2	0.0605	1080	0.2705	1824	0.135	1335
In_expB2	0.0657	1259	**0.2727**	**1826**	0.1537	1581
In_expC2	0.0700	1288	0.2704	1826	0.1551	1609
InL2	0.0629	1259	0.2575	1826	0.1521	1570
PL2	0.0730	1295	0.2510	1803	0.1575	1658
LemurTF_IDF	0.0517	1146	0.2269	1814	0.1319	1425
TF_IDF	0.0651	1302	0.2525	1818	0.1452	1627
nnc_ntc	**0.0779**	**1270**	0.2190	1760	**0.161**	**1698**
ntc_ntc	0.0630	1235	0.2154	1776	0.1525	1623
lnc_ntc	0.0722	1269	0.2270	1784	0.1585	1667
ntn_ntn	0.0649	1250	0.2140	1792	0.1464	1643
lnn_ntn	0.0658	1284	0.2346	1789	0.1527	1684
ltn_ntn	0.0512	1166	0.2167	1785	0.1297	1511
lsn_ntn	0.0426	1028	0.1856	1787	0.1140	1376
CombSUM	0.0691	1314	0.2637	1844	0.1635	1680
WCombSUM	0.0692	1315	0.2645	1847	0.1631	1680
WHCombSUM	0.0857	1330	0.2803	1894	0.1771	1733

We can compare our heuristic method with the results of other IR systems on the same test set (using the 33 English test queries and the automatic transcripts – the required run for the CLSR task at CLEF 2007). For this setting we obtained a MAP score of 0.0857. This result was approximately the same as the best system proposed by [157] (the MAP score was 0.855). It can be considered better because that system had a drawback with regard to the running time. It took a long time to run 15 weighting schemes for each query, and then to fuse the results, but our fusion methods select at the most 6 retrieval strategies from the 20 retrievals strategies for the experiments conducted on Auto, Manual, or Auto+Manual segment representations . Moreover, our method is better than the other 4 systems that participated in the task [158], as reported in Table 27.

Table 27: Results for our heuristic method (WHCombSUM) and the 5 teams that participated in the CLSR task at CLEF 2007, on the English test queries.

Submitted run	MAP score
WHCombSUM	0.0857
UO	0.0855
DCU	0.0787
BLLIP	0.0785
UC	0.0571
UVA	0.0444

8.5. Conclusion

In this chapter, we explored the idea of combining the results of different retrieval strategies and how to train the weights for each retrieval strategy in a systematic and novel way. Our experiments results showed that our method is better than the classical data fusion technique *CombSUM* and *WCombSUM.*

Chapter 9. Class-Based Fusion

9.1. Introduction

In this chapter, we address the issue of high variation among the retrieval strategies or document representations which affect the combination of their outputs. Lee [109] analyzed the overlap values of result sets from six different participants in TREC-3; he found that low overlap in non-relevant and high overlap in relevant documents is critical to improving effectiveness. We believe that the data fusion method should be able to combine the results that have high retrieval effectiveness with the results that have low retrieval effectiveness. Therefore, we propose a novel data fusion technique to fuse the results of different document representations, where the quality of the retrieval results varies from low to high quality.

Our investigation on the MALACH speech collection, where different segment representations are available, shows that neither the classical data fusion (*CombSUM*), nor the weighted version (*WCombSum*) can improve the retrieval. We propose a novel class-based data fusion technique to deal with this issue, where the retrieved segments – from each document representation involved in the fusion - are classified according to the quality of each segment, into three classes: high, intermediate, and low quality class, and then the similarity scores of each segment are fused using the classical *CombSUM*.

The remainder of this chapter is organized as follows. Section 9.2 is pointing out directly-relevant related work. Section 9.3 describes the data fusion technique proposed in this chapter. Section 9.5 presents our experimental results. Section 9.6 presents conclusions.

9.2. Related Work

In the literature, there are three studies that tackled the problem of fusing different retrieval strategies or different document representations with high variance.

He and Ahn [118] investigate the fusion of the retrieval results of different segment representations in MALACH collection, where the retrieval results are varied among the representations. Their

first fusion method was based on the classical WCombMNZ method, where a predefined weight for each representation is calculated based on four effectiveness measures (MAP, R-PREC, P10, and Avg-Recall) on the training data. There experiments showed that the four ways of choosing the weights is better than the classical CombMNZ, but is still significantly lower than the best system involved in the fusion. The second part of their work was proposing multiple iteration data fusion technique, where they fused the results of the four ways to select the weights and the best representation results from the first iteration. In the second iteration, they tried different values to weight each retrieval results on the training data. There results showed that multi-level data fusion is better than the simple fusion technique and better than the best system involved in the fusion.

The second study, conducted by He and Wu [163], investigates a data fusion technique where the retrieval results involved in fusion are diverse. He tried to fuse the results with low, high, or a combination of low and high. They applied their fusion technique to different segment representations of MALACH spoken document collection, and to HARD collection from TREC. Their method combines the normalized scores of a document with a predefined weight for each ranked list based on MAP for each rank-list on the training data, then multiplying the final score of each document by the summation of all ranked-list's weights if it the document appeared in these ranked-lists. In MALACH collection the retrieval results are varied among various representations. In HARD collection, they chose different runs submitted to TREC, where the results are varied among various runs. When the combined results in similar quality, their results showed that the new methods is better than the classical data fusion technique (CombMNZ), but when the combined results are diverse in the quality, their fusion methods is butter than CombMNZ, but is not significantly better than the best results involved in the fusion.

The last work was done by Jones et al [122], where a combination of different segment representations was conducted during the indexing phase to produced new index, where a weight was assigned to each segment representation during the indexing. Their indexing technique was an extended version of BM25 called BM25F. Their results showed that BM25F could produce improved search accuracy.

In our experiments, we will use CombSUM and WCombSUM as baseline method, to compare it to our new technique. As a base case, we will consider the MAP scores as the weights in the training phase for WCombSUM.

9.3. Class-Based Fusion

In this chapter we will discuss the case when we have different retrieval strategies and there are large differences in the effectiveness (significant difference), or we have one retrieval strategy and different representations for the documents, so that when we apply the retrieval strategy to the different representations, there are a significant differences among the different representations. Because of these differences, the basic fusion methods fail to improve the retrieval due to the noises from bad strategies or representations.

For example, the MALACH test collection contains 8104 segments from 272 interviews with Holocaust survivors and each segment contains different versions of automatic transcriptions, two sets of automatically-generated thesaurus terms, manually generated summaries, and manually-generated thesaurus terms. Each of them can be viewed as a representation for the segment. The first representation is when we index the automatically-generated data (Auto). The second one is when we index the manually-generated data (Manual), and the third one is when we index the automatic and the manually-generated data together (Auto+Manual). If we apply any retrieval strategy to each representation, there are big differences among the representations, for example as shown in Table 28, the MAP score for Auto, Manual, and Auto+Manual are 0.1041 , 0.3321, and 0.2837, respectively. The basic fusion methods like CombSUM or WCombSUM where the weights are the MAP scores on training data, the MAP scores for the fusion methods are 0.2844 and 0.3272, respectively.

We are looking for a fusion technique that can handle the variations among the retrieval strategies or the document representations.

Table 28. The retrieval results on MALACH collection using the weighting scheme DLH13 from the Terrier IR system on training data.

	Training	**Test**
Auto	0.1041	0.0735
Manual	0.3321	0.2560
Auto+Manual	0.2837	0.1606
CombSUM	0.2844	0.1953
WCombSUM	0.3272	0.2393

To achieve this goal, we will divide the retrieved documents from all the retrieval strategies or the document representations into three classes: the first one is expected to have the best precision values, the second one has intermediate precision values, and the last one has low precision; we will call these classes high, intermediate, and low class, respectively. Since the Manual experiment has the best MAP, we will assume the high class will have the top n documents from the Manual experiment. The intermediate class will have the next m documents from Manual and the top m documents from Auto+Manual. Finally, the low class will have the remaining documents from Manual, Auto+Manual, and all the documents from Auto experiment. Note that the intersection between the three classes has to be mutually exclusive, i.e., if a document d appears in the top n documents from Manual and in the top m documents from Auto+Manual, d will be included in the high class, not in the intermediate class.

The next step is how to estimate the values for n and m? Using the evaluation of the three experiments on training data; for this stage we will choose interpolated precision at 11-level recall levels. To estimate n, which represents the separation point between the high class and the intermediate class we, will chose the maximum precision on auto-manual experiment, then find the level of recall that represent this value in manual experiment, which is actually the same as looking at the length of the document list at the cut-off point; finally, we multiply this recall level by 1000 to calculate n (since the number of retrieved documents for each retrieval strategy is 1000, we take a portion of this number, which is proportional to the recall level). We do the same procedure for m; chose the maximum precision on the Auto experiment, then find the level of recall on Manual+Auto and multiply it by 1000.

For example, Table 29 represents the 11-levels of recall-precision for the three experiments mentioned in Table 28. To estimate n, first we have to find the best precision in Auto+Manual, which is 0.697; then we have to find the level of recall that represents this value in the Manual experiment (0.1), and finally multiply this recall level by 1000; therefore, the estimated value for n is 100. We do the same thing for m; the maximum precision value in Auto is 0.424; the level of recall that represents this value according to the evaluation of the Auto+Manual experiment is 0.3; so, m is equal to 300. The high class will contain the top 100 documents from Manual; the intermediate class will contain the next 300 documents from Manual and the top 300 from Auto+Manual; finally, the low class will contain the remaining documents from Manual and Auto+Manual (600 and 700, respectively) and all the documents from Auto that were not included neither in the high class nor in

the intermediate class. The three classes are mutually exclusive. In the above example, if one of the top 100 documents from Manual happens to be in the set of top 300 from Auto+Manual, then this document will be in the high class, not the intermediate one.

Table 29: 11-level interpolated recall-precision values for the three experiments: Manual, Auto+Manual, and Auto. We show how to derive n and m, as explained in the text.

	Manual	Auto+Manual	Auto
Recall	**Precision**	**Precision**	**Precision**
0%	0.722	**0.697**	**0.424**
n=100 10%	0.577	0.504	0.247
20%	0.507	0.439	0.189
m=300 30%	0.435	0.353	0.146
40%	0.405	0.315	0.115
50%	0.353	0.282	0.091
60%	0.301	0.256	0.061
70%	0.242	0.200	0.041
80%	0.154	0.152	0.017
90%	0.090	0.088	0.023
100%	0.032	0.025	0.001

The final step is to fuse the similarity scores of each document and to sort them in decreasing order in each class separately, then arrange the documents for the high class first, then the intermediate class, and finally the low class. To fuse the similarity scores, we could use CombSUM or WCombSUM. We have to normalize the similarity scores according to the maximum and minimum in each class. In our experiments, for any run that uses the class-based fusion, we will use the prefix "WC" before the method name, i.e., WCCombSUM.

9.4. Experimental Results

The candidate retrieval strategies (weighting schemes) for our fusion system were provided by two IR systems: SMART [49, 156] and Terrier [62, 64] . We used the (nnc.ntc, ntc.ntc, lnc.ntc, ntn.ntn, lnn.ntn, ltn.ntn, lsn.ntn) weighting schemes from SMART [49, 156], and (BB2, BM25, DFR_BM25, DFRee, DLH13, DLH, IFB2, In_expB2, In_expC2, InL2, PL2, LemurTF_IDF, and TF_IDF); for more detail about these technique see chapter 2.

We conducted three types of experiments, based on the fields which were indexed. In the first one, the automatic transcripts (ASRTEXT2006B), and two automatic keywords (AK1 and AK2)

were used for indexing the documents; we call this experiment Auto. In the second experiment, we indexed the manual keywords and the manual summaries for each document; we named this experiment Manual. In the last experiment we indexed the automatic transcripts, the two automatic keywords fields, the manual summaries, and the manual keywords, we call this experiment Auto+Manual. The title and description fields from each topic are used as query. Table 30 shows some statistics about each experiment.

One interesting observation is that the number of terms (distinct words) in the manual fields is about half of the number of terms in the automatic fields. The number of tokens (total number of words) in the manual fields is about 16% of the number of tokens in the automatic fields. The average term frequencies are 39, 125, and 125 for Manual, Auto, and Auto+Manual, respectively. This ratio is very high: about four times more in the Auto fields. We also note that combining Auto and Manual brings about 14% of the terms to the Auto+Manual list of terms, which means that there is more information in the combined fields.

Table 30: Some statistics about the number of terms and the number of tokens for the three experiments.

	Number of terms	Number of tokens
Auto	13,605	1,711,684
Manual	7,131	278,717
Auto + Manual	15,884	1,990,401

9.4.1 Manual Summaries and Keywords versus Automatic Transcripts

Experiments on manual keywords and manual summaries (Manual) available in the test collection showed high improvements over automatic transcripts and automatic keywords (Auto). The MAP score jumped from 0.0779 to 0.2727 on the test data. Also, if we indexed the Manual fields and the Automatic fields together (Auto+Manual), the MAP score jumped to 0.161, but it is far from the results on the Manual. This was also the case in the systems that participated in CLEF-CLSR. We are looking for a justification of why the difference is so big between the results of the Auto experiment and the Manual experiment, and why when we merge the Auto with Manual we do not reach the performance of the Manual fields. Since there are no manual transcripts available for the segments, we cannot know how the word error rate (WER) affects the retrieval.

In our view there are four factors that may affect the retrieval. The first factor is related to the nature of the summary and manual keywords; these fields were generated by experts, for example

the manual summary is three-sentences long on average and answers four main questions: who? what? when? and where?. So, the summary is a very concise representation of the segments.

The second factor is how the automatic transcript or the manual summary covers the search terms from the training and test topics. To find out the effect of this factor we count the missing terms for each experiment in the training and test topics for title and description field. The results are shown in Table 31. We noticed that the number of the missing terms is approximately the same for Manual and Auto, and for Auto+Manual is approximately half the missing number of terms from Manual or Auto. Therefore, we cannot consider the missing term as the factor which affects the large difference in MAP score between Auto and Manual.

The third factor could be related to the ability of the search terms to discriminate among the documents. The classic discrimination measure is the idf value for the search terms. Therefore we compute the average idf for the training and test topics; the values are shown in Table 31. We can see that the average idf for Auto and Auto+Manual is less than for Manual. So, the topics ability to discriminate the documents in the Manual experiments is higher than for Auto or Auto+Manual.

The fourth factor is the average term frequency. It is much larger in Auto and Auto+Manual (125) than in Manual (39), as previously concluded from Table 30.

Since the manual summaries and the automatic transcripts complement each other, each one brings new terms to the document structure as shown also in Table 29. Mixing the two fields is supposed to improve the retrieval, in theory. From the results, it is clear that simple merging technique - during the indexing - does not help. A better way to combine or fuse the two fields during the indexing was addressed by [122].

In the next section, we will presents the experiments results for the class-based fusion which improve the retrieval, and benefit from the different information included in different segment representation.

Table 31: The average idf values, and number of missing search terms from title and description fields, for training (681 terms) and test (356 terms) topics

	IDF Training	**IDF Test**	**Missing Training**	**Missing Test**
Auto	1.22	1.08	28	8
Manual	1.75	1.74	27	9
Auto+Manual	1.22	1.05	10	5

9.4.2 Class-Based Fusion Experiments

We have applied our class-based fusion proposed in section 4.5 to fuse the results from the three segments representations Auto, Manual, and Auto+Manual for each retrieval strategy (weighting scheme) from SMART or Terrier.

The baselines are the best retrieval run (the results from the Manual representation run) and the classical retrieval *WCombSUM*, where the weights in *WCombSUM* are represented by the MAP of each run on training data.

As shown in Table 32 and Figure 29, the classical fusion technique (*WCombSUM*) does not improve the results comparing to the best run involved in the fusion process for the 20 retrieval strategies; but our method was better than the best run involved in the fusion for all the 20 retrieval strategy. For 15 out of 20 runs, the improvement was significant, based on a one-tailed Wilcoxon signed rank test with ($p < 0.05$). Also, our method was significantly better than the classical WCombSUM for all the 20 retrieval strategies.

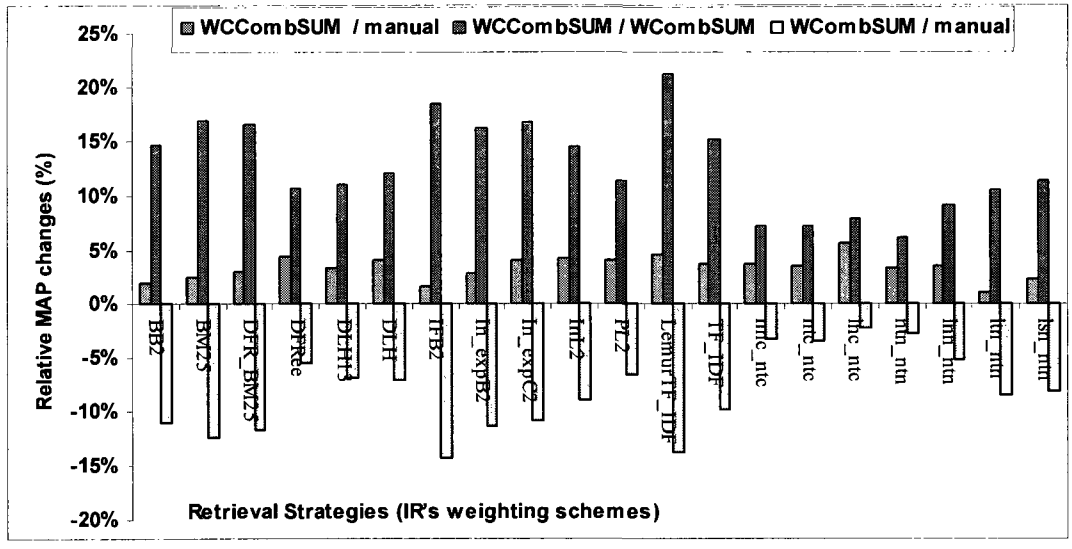

Figure 29: Relative MAP changes between WCCombSUM and the best pre-fusion run (manual representation), WCCombSUM and WCombSUM, and WCombSUM and the best pre-fusion run (manual representation). The fusion is applied to 20 retrieval strategies from SMART and Terrier to fuse 3 segment representations: Auto, Manual, Auto+Manual.

We conclude from our experiments that the information in meta-data like manual summaries and keywords complement the information contained automatic transcriptions and automatic key-

words, and we could benefit from this feature to post-fuse the results of each representation and improve the retrieval.

Table 32: Results (MAP scores, and number of relevant documents retrieved) for 20 weighting schemes from Smart and Terrier, and the results of the fusion methods (WCombSUM and WCCombSUM), on the test data.

Weighting scheme	Auto		Manual		Auto+Manual		WCombSUM		WCCombSUM	
	MAP	Rel-Ret	MAP	Rel.-Ret	MAP	Rel-Ret	MAP	Rel-Ret	MAP	Rel-Ret
BB2	0.0441	972	0.2699	1826	0.0970	1133	0.2402	1869	0.2752	1888
BM25	0.0567	1120	0.2490	1824	0.1404	1381	0.2182	1874	0.2551	1882
DFR_BM25	0.0580	1122	0.2558	1818	0.1408	1407	0.2261	1878	0.2635	1889
DFRee	0.0695	1298	0.2527	1822	0.1586	1697	0.2387	1897	0.2640	1900
DLH13	0.0735	1335	0.2560	1825	0.1606	1720	0.2384	1898	0.2647	1890
DLH	0.0719	1325	0.2460	1812	0.1606	1707	0.2287	1875	0.2560	1878
IFB2	0.0605	1080	0.2705	1824	0.135	1335	0.2320	1899	0.2747	1900
In_expB2	0.0657	1259	0.2727	1826	0.1537	1581	0.2416	1918	0.2805	1925
In_expC2	0.0700	1288	0.2704	1826	0.1551	1609	0.2409	1915	0.2812	1911
InL2	0.0629	1259	0.2575	1826	0.1521	1570	0.2346	1898	0.2685	1898
PL2	0.0730	1295	0.2510	1803	0.1575	1658	0.2347	1876	0.2613	1873
Lemur TF_IDF	0.0517	1146	0.2269	1814	0.1319	1425	0.1956	1867	0.2371	1874
TF_IDF	0.0651	1302	0.2525	1818	0.1452	1627	0.2277	1890	0.2620	1884
nnc_ntc	0.0779	1270	0.2190	1760	0.161	1698	0.2119	1837	0.2271	1845
ntc_ntc	0.0630	1235	0.2154	1776	0.1525	1623	0.2080	1845	0.2230	1873
lnc_ntc	0.0722	1269	0.2270	1784	0.1585	1667	0.2222	1865	0.2398	1879
ntn_ntn	0.0649	1250	0.2140	1792	0.1464	1643	0.2084	1857	0.2212	1867
lnn_ntn	0.0658	1284	0.2346	1789	0.1527	1684	0.2226	1880	0.2429	1897
ltn_ntn	0.0512	1166	0.2167	1785	0.1297	1511	0.1984	1844	0.2191	1880
lsn_ntn	0.0426	1028	0.1856	1787	0.1140	1376	0.1706	1795	0.1899	1832

9.5. Conclusion

We have addressed the case when there are large differences in the effectiveness between the retrieval strategies or the document representations involved in the fusion, where classical techniques failed badly to improve the results. The solution was a class based method.

Finally, we have showed that meta-data complemented the error-full transcription, and we could benefit from the class-based fusion to improve the retrieval.

Chapter 10. Conclusions

This chapter reviews the contributions of the thesis and discusses further work arising from several issues encountered along the way.

10.1. Contributions

We obtained the best retrieval results among all the teams that participated in this track. We believe that the improved performance is due to the choice of the weighting schemes used for indexing the document and query terms, relevance feedback approaches, and the data fusion methods.

We have investigated the hypothesis that the phonetic form could help compensate for the speech recognition errors made when the collection was produced. The results showed that 4-gram phoneme for indexing did not improve the retrieval compared to word-based indexing, but when combining phonetic and text forms (by simply indexing both phonetic n-grams and text), the result improved compared to using only the phonetic forms. Nonetheless, the MAP scores were lower than the results on the text form for documents and queries.

The idea of using multiple translations proves to be good. More variety in the translations would be beneficial. The online MT systems that we used are rule-based system. Adding translations by statistical MT tools might help, since they produce radically different translations.

We experimented with two different systems: Terrier and SMART, with various weighting scheme for indexing the document and query terms. We proposed two new approaches for query expansion that use collocations with high log-likelihood ratio, a thesaurus-based approach; and we used two other approaches form Terrier: the Kullback-Leibler (KL) model and the Bose-Einstein model. Relevance feedback methods produced only small improvements. So, query expansion methods do not seem to help a lot for this collection.

The improvements of mean word error rates in the ASR transcripts (of ASRTEXT2006A relative to ASRTEXT2004A) did not improve the retrieval results. Also, combining different ASR tran-

scripts (with different error rates) did not help.

On the manual data, the best MAP score that we obtained is 32%, for English topics. On automatic data the best result is 9% MAP score. We think that the justification for the difference is due to the fact that the manual summaries contain different words to represent the content of the speech segments. Another reason is that the poor quality of the ASR transcripts severely hurts the performance of IR systems, for this collection.

For some experiments, Terrier was better than SMART, for other it was not; therefore we cannot clearly choose one or another IR system for this collection. For that reason, we proposed the data fusion technique to improve the retrieval or at least to compensate for the variation among the systems.

We proposed five methods to combine different weighting schemes (retrieval strategies) from different systems:

1. MAP-Recall weights for model fusion technique based on a weighted summation of normalized similarity measures; the weight for each scheme was based on the relative precision and recall on the training data.
2. Cluster–based model fusion (WRCombMNZ): This technique is based on clustering the training topics according to their tf-idf (term frequency-inverse document frequency) properties, and selecting the best models for each cluster. When the system runs on a test topic, the cluster of the topic needs to be determined and the combination of models of this cluster is used.
3. Heuristic-based weights for model fusion (WCcombSUM): This is a novel technique to select the retrieval strategies to be involved in the fusion and assigning a weight for each one.
4. Class-based model fusion (WCCombSUM): This technique addresses the issue of high variation among the retrieval strategies or document representations which affect the combination of their outputs. In this technique, the retrieved segments – from each document representation involved in the fusion - are classified according to the quality of each segment to three classes: high, intermediate, and low quality class, and then the similarity scores of each segment are fused using the classical CombSUM.
5. Probability-based model fusion (CombMultPROB and CombTotPROB): This fusion technique attempt to derive the fusion based on the probability theory, inspired from the

very popular experiment in the probability drawing balls for boxes.

The first three fusion methods have the same performance in term of MAP score, and the difference was not significant. The methods differ in terms of the number of the retrieval strategies involved in the fusion process and in the way we select these strategies. In the first one, we select all the retrieval strategies and assing reasonable weight to each retrieval strategy based on the recall and MAP score. This technique has a shortcoming in terms of time; it needs longer time as the number of retrieval strategies is higher. In the second and third fusion methods, we have proposed different ways to select the retrieval strategies to be involved in the fusion. In this way, we run fewer strategies, and we saved time. The class-based fusion proves to be effective when there is a high variation among the retrieval strategies (it should not be applied if this in not the case). Finally, we could use the probability-based model fusion in any of the previous techniques. The strengths of the probabilistic technique is a theoretical one: we derived the fusion formula based on basic laws from the probability theory.

In general, data fusion helps to improve the retrieval; the difference is statistically significant for some experiments comparing to the best retrieval strategy involved in the fusion and all the differences are statically significant comparing to the classical technique in data fusion (CombSUM and WCombSUM).

Combining query expansion methods and data fusion helped to improve the retrieval significantly comparing to the median and average of all the required runs submitted by all the teams that participated in CLEF-CLSR 2007.

10.2. Our System vs. Other Systems that Participated in the CLEF-CLSR Track

The CL-SR track at CLEF was run for three consecutive years 2005, 2006, and 2007. We have participated in the CL-SR track at CLEF for each of the three years. Our system achieved very successful results comparative to the other systems. In the first year, we focused on the investigation of different weighting schemes from the SMART information retrieval system, and combining seven tools for translating the queries for the cross-language retrieval task. In the second year, we have developed the relevance feedback model based on the collocation lists extracted from ASR transcripts, and investigating another query expansion methods implemented in the Terrier information retrieval system (especially the Kullback-Leibler (KL) model for query expansion). In the last year, we have

proposed the thesaurus-based query expansion method, and we investigated another method based on Bose-Einstein model for query expansion. The main contributions of the 2007 submission consisted in combining the previous methods: we developed the two model fusions techniques to combine the results from 15 weighting schemes.

The experiments covered three areas, the first one when using the manual fields (summaries and manual keywords) for indexing the documents (we called it Manual English). The second area when using just the automatic fields for indexing the documents (ASR transcripts and automatic keywords) and using the title and the description from the topics, this run is the required run and we called it Auto-English. The cross-language experiments is the last area, where the automatic fields are used from the documents and French, Spanish, German, or Czech topics are used; we call them Auto-French, Auto-German, Auto-Spanish, or auto-Czech respectively.

Our system achieved the best results among all systems that participated in CL-SR track in 2005 and 2007. In 2006, we have achieved the second-best results for Auto-English (the required run), and for Auto-French (optional run). Figures 30, 31, and 32 show the results for all the systems that participated in the CL-SR track for 2005, 2006, and 2007, respectively. For brief descriptions of the systems that participated in CLEF-CLSR, see Section 4.1.

In 2005 the success of our system was due to the careful selection of the appropriate weighting scheme or retrieval strategy from SMART information retrieval system. SMART has a variety of weighting schemes based on the vector space model. We have selected the best strategy after applying it to training data. The reader could refer to [8] for more details.

In 2006 we have proposed different query expansion techniques: one is based of adding terms that have high log-likelihood ratio to the original query, and the second one is based on adding terms from a thesaurus. We have tried to combine these approaches with the relevance feedback approach implemented in Terrier information retrieval system. The reader could refer to [9] for more details.

In 2007 Data Fusion was the core of our research. We combined the results of different retrieval strategies by attaching weights to each one, based on their performance on the training data. The reader could refer to [157] for more details.

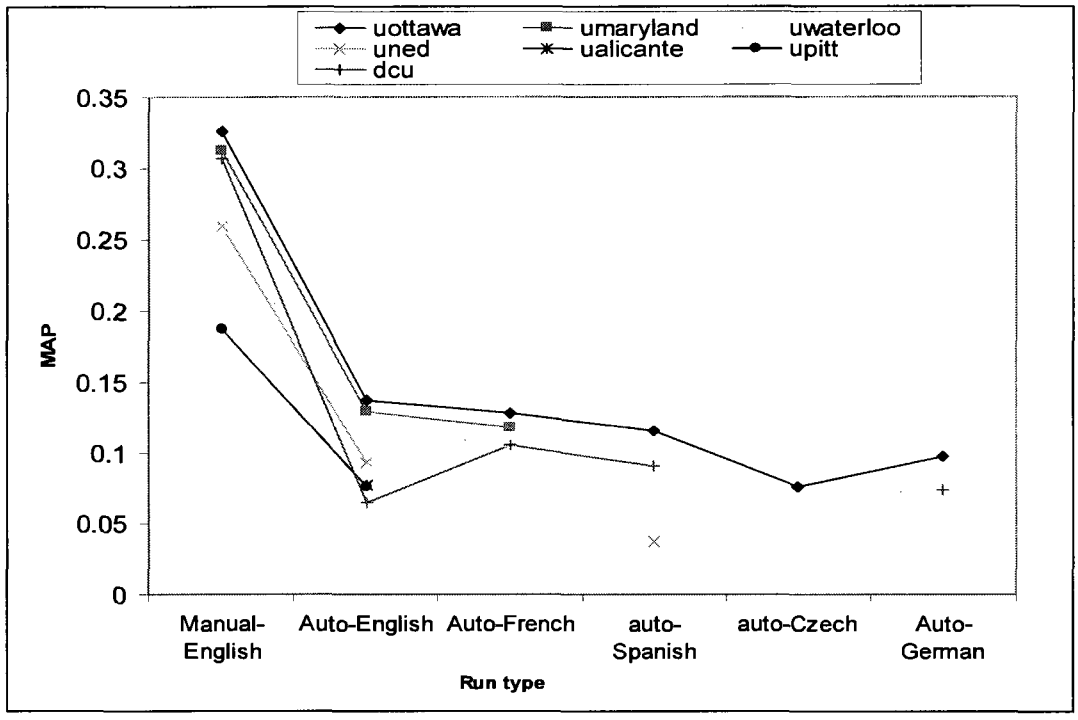

Figure 30: Results for all the systems that participated in CLEF-CLSR 2005, for test data.

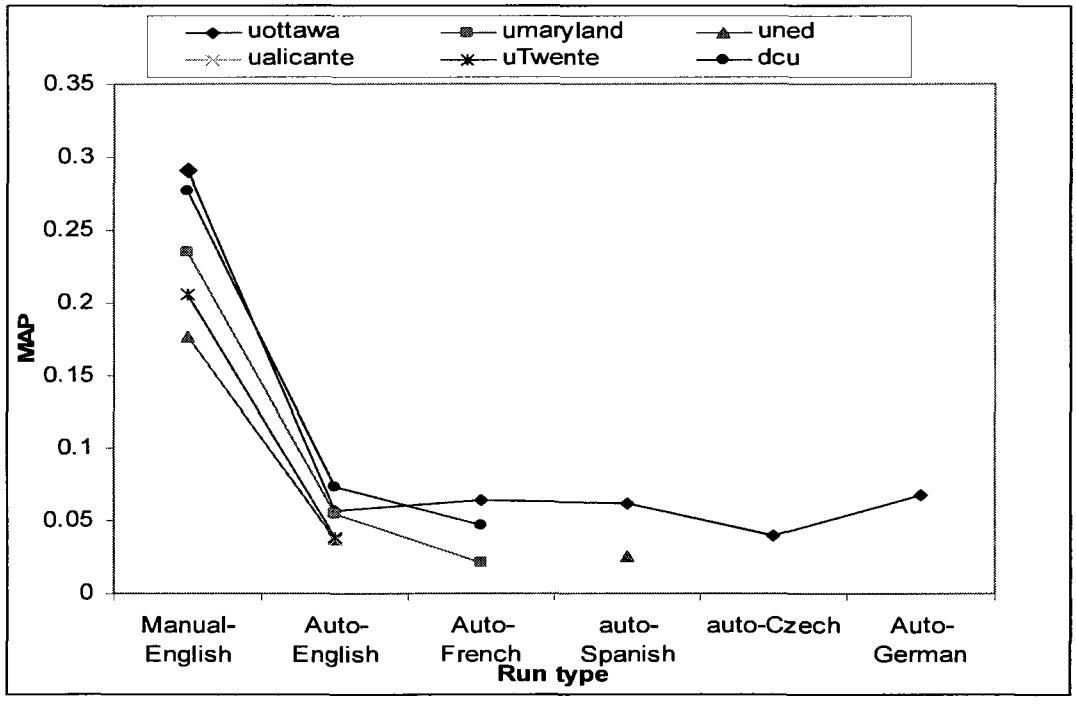

Figure 31: Results for all participants on the CLEF-CLSR 2006 test collection.

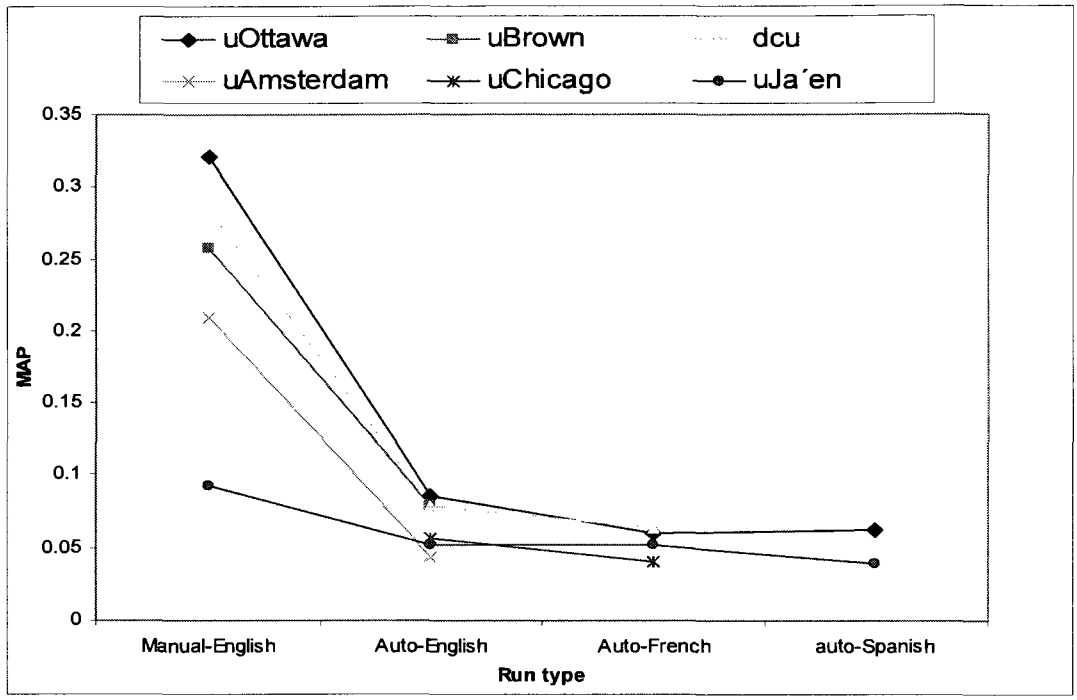

Figure 32: Results for all the systems that participated in CLEF-CLSR 2007.

10.3. Future Works

The main problem in speech retrieval is word mismatch due to misrecognized words, because the automatic transcription may not contain all the words that were actually spoken (word recall problem) or may contain words that were not spoken (word precision problem). Singhal et al. argue that IR would benefit from high word-recall, and that it would be less influenced by poor word precision [20].

There are several ways in the literature to address the word mismatch problem. One way is to expand the query using pseudo relevance feedback or thesaurus-based expansion. Another way is to expand the documents: transcribed segments are expanded with words from textual documents that are topically related to the segments. Document expansion would be effective if some of the added words are already in the transcription; this would give these higher weights words and would reduce the importance of the other words that may not have been spoken. On another side, it could add to the transcription new words which might have been spoken, but the ASR failed to recognize them.

Document expansion might be beneficial if the side collection or the expansion corpus is related the speech collection [150]. One of the key successes in TREC-SDR track was documents expansion [17-26]. However, previous work on applying document expansion on spontaneous speech collection showed that document expansion helps to improve the retrieval, but the improvement was not significant [116].

As a part of our plans to improve the retrieval from spontaneous speech collection, we are planning to investigate document expansion from side collection techniques for MALACH collection.

Other directions for future work are to investigate more methods for data fusion, and to investigate more methods to compensate for the speech recognition errors in the ASR transcripts.

Appendix A: SMART Term Weighting Scheme Codes

Table 33 Some of SMART's weighting schemes that are used for our experiments.

	Weighting scheme for document or query	Formula
1	mpc	$W_{mpc} = \frac{\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N-df}{df}\right)}{\sqrt{\sum_m \left[\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N-df}{df}\right)\right]^2}}$ (11.1)
2	mts	$W_{mts} = \frac{\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N}{df}\right)}{\sqrt{\sum_m \left[\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N}{df}\right)\right]}}$ (11.2)
3	nts	$W_{nts} = \frac{(tf) * \left(\log \frac{N}{df}\right)}{\sqrt{\sum_m \left[(tf) * \left(\log \frac{N}{df}\right)\right]}}$ (11.3)
4	ntn	$W_{ntn} = (tf) * \left(\log \frac{N}{df}\right)$ (11.4)
5	npc	$W_{npc} = \frac{(tf) * \left(\log \frac{N-df}{df}\right)}{\sqrt{\sum_m \left[(tf) * \left(\log \frac{N-df}{df}\right)\right]^2}}$ (11.5)
6	mtc	$W_{mtc} = \frac{\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N}{df}\right)}{\sqrt{\sum_m \left[\left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N}{df}\right)\right]^2}}$ (11.6)

7	ntc	$W_{npc} = \frac{(tf) * \left(\log \frac{N}{df}\right)}{\sqrt{\sum_m \left[(tf) * \left(\log \frac{N}{df}\right)\right]^2}}$ (11.7)
8	mtn	$W_{mtn} = \left(\frac{tf}{\max\_tf}\right) * \left(\log \frac{N}{df}\right)$ (11.8)
9	npn	$W_{npn} = (tf) * \left(\log \frac{N-df}{df}\right)$ (11.9)
10	lsn	$W_{lsn} = \left(\ln(tf)+1\right) * \left(\log \frac{N}{df}\right)^2$ (11.10)
11	atn	$W_{atn} = \left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right) * \left(\log \frac{N}{df}\right)$ (11.11)
12	asn	$W_{asn} = \left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right) * \left(\log \frac{N}{df}\right)^2$ (11.12)
13	snn	$W_{snn} = (tf)^2$ (11.13)
14	sps	$W_{nts} = \frac{(tf)^2 * \left(\log \frac{N-df}{df}\right)}{\sqrt{\sum_m \left[(tf)^2 * \left(\log \frac{N-df}{df}\right)\right]}}$ (11.14)
15	nps	$W_{nts} = \frac{(tf) * \left(\log \frac{N-df}{df}\right)}{\sqrt{\sum_m \left[(tf) * \left(\log \frac{N-df}{df}\right)\right]}}$ (11.15)
16	atc	$W_{atc} = \frac{\left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right) * \left(\log \frac{N}{df}\right)}{\sqrt{\sum_m \left[\left(0.5+0.5\left(\frac{tf}{\max\_tf}\right)\right) * \left(\log \frac{N}{df}\right)\right]^2}}$ (11.16)

References

[1] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees, "The TREC Spoken Document Retrieval Track: A Success Story," in *Proceedings of the RIAO: Content-Based Multimedia Information Access* PARIS, 2000.

[2] A. James, "Perspectives on Information Retrieval and Speech," *Lecture Notes in Computer Science: Information Retrieval Techniques for Speech Applications* vol. 2273, 2002.

[3] J. S. Garofolo, E. M. Voorhees, V. M. Stanford, and K. S. Jones, "TREC-6 1997 Spoken Document Retrieval Track Overview and Results," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[4] J. S. Garofolo, E. M. Voorhees, C. G. P. Auzanne, V. M. Stanford, and B. A. Lund, "1998 TREC-7 Spoken Document Retrieval Track Overview and Results," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[5] R. W. White, D. W. Oard, G. J. F. Jones, D. Soergel, and X. Huang, "Overview of the CLEF-2005 Cross-Language Speech Retrieval Track " *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 744-759, 2006.

[6] D. W. Oard, J. Wang, G. J. F. Jones, R. W. White, P. Pecina, D. Soergel, X. Huang, and I. Shafran, "Overview of the CLEF-2006 Cross-Language Speech Retrieval Track," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 744-758, 2007.

[7] P. Pecina, P. Hoffmannov´a, G. J. F. Jones, Y. Zhang, and D. W. Oard, "Overview of the CLEF-2007 Cross Language Speech Retrieval Track," *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* vol. 5152, pp. 674-686, 2008.

[8] D. Inkpen, M. Alzghool, and A. Islam, "Using Various Indexing Schemes and Multiple Translations in the CL-SR Task at CLEF 2005," *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 760–768, 2006.

[9] M. Alzghool and D. Inkpen, "Experiments for the Cross Language Speech Retrieval Task at CLEF 2006," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 778-785, 2007.

[10] M. Alzghool and D. Inkpen, "Model Fusion Experiments for the CLSR Task at CLEF 2007," *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* vol. 5152, pp. 695-702, 2008.

[11] D. W. Oard, D. Soergel, D. Doermann, X. Huang, G. C. Murray, J. Wang, B. Ramabhadran, M. Franz, and S. Gustman, "Building an information retrieval test collection for spontaneous conversational speech," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* Sheffield, United Kingdom: ACM, 2004.

[12] E. A. Fox and J. A. Shaw, "Combination of Multiple Searches," in *Proceedings of the Second Text REtrieval Conference (TREC 2)* Gaithersburg, Maryland, USA: National Institute of Standards and Technology (NIST), 1993.

[13] B. T. Bartell, G. W. Cottrell, and R. K. Belew, "Automatic Combination of Multiple Ranked Retrieval Systems," in *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* Dublin,Ireland: ACM/Springer, 1994.

[14] C. C. Vogt and G. W. Cottrell, "Fusion Via a Linear Combination of Scores," *Information Retrieval* vol. 1, pp. 151-173, 1999.

[15] G. J. F. Jones, J. T. Foote, K. Sp, J. rck, and S. J. Young, "Retrieving spoken documents by combining multiple index sources," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* Zurich, Switzerland: ACM, 1996.

[16] K. Ng, "Information fusion for spoken document retrieval," in *Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference - Volume 04* Istanbul, Turkey: IEEE Computer Society, 2000.

[17] J. Allan, J. Callan, W. B. Croft, L. Ballesteros, R. Swan, and J. Xu, "INQUERY Does Battle With TREC-6," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[18] D. Abberley, S. Renals, G. Cook, and T. Robinson, "Retrieval of Broadcast News Documents With The THISL System," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[19] J. Allan, J. Callan, M. Sanderson, J. Xu, and S. Wegmann, "INQUERY and TREC-7," in *Proceedings of the Seventh Text REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[20] A. Singhal, J. Choi, D. Hindle, D. D. Lewis, and F. Pereira, "AT&T at TREC-7," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[21] D. Abberley, S. Renals, D. Ellis, and T. Robinson, "The THISL SDR System At TREC-8," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[22] J. Allan, J. Callan, F.-F. Feng, and D. Malin, "INQUERY and TREC-8," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[23] J.-L. Gauvain, Y. d. Kercadio, L. Lamel, and G. Adda, "The LIMSI SDR System for TREC-8," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[24] S. E. Johnson, P. Jourlin, K. S. Jones, and P. C. Woodland, "Spoken Document Retrieval for TREC-8 at Cambridge University," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[25] C. Lundquist, O. Frieder, D. Holmes, and D. Grossman, "A parallel relational database management system approach to relevance feedback in information retrieval," *Journal of the American Society for Information Science* vol. 50, pp. 413-426, 1999.

[26] A. Singhal, S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira, "AT&T at TREC-8," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[27] D. Inkpen, M. Alzghool, G. Jones, and D. Oard, "Investigating Cross-Language Speech Retrieval for a Spontaneous Conversational Speech Collection," in *Proceedings of the Human Language Technology Conference / North American chapter of the Association for Computational Linguistics, HLT-NAACL 2006,* New York City, NY, 2006.

[28] M. Alzghool and D. Inkpen, "Model Fusion Experiments for the Cross Language Speech Retrieval Task at CLEF 2007," *Lecture Notes In Computer Science : Advances in Multilingual and Multimodal Information Retrieval,* vol. 5152/2008, pp. 695-702, 2008.

[29] M. Alzghool and D. Inkpen, "Clustering the topics using TF-IDF for model fusion," in *Proceeding of the 2nd PhD workshop on Information and knowledge management* Napa Valley, California, USA: ACM, 2008.

[30] M. Alzghool and D. Inkpen, "Combining Multiple Models for Speech Information Retrieval," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)* Marrakech, Morocco, 2008.

[31] M. Alzghool and D. Inkpen, "Cluster-based Model Fusion for Spontaneous Speech Retrieval," in *Proceedings of the Workshop on Searching Spontaneous Conversational Speech, SIGIR 2008* Singapore, 2008.

[32] M. Alzghool and D. Inkpen, "Exploring Fusion in a Spontaneous Speech Retrieval Task," in *Proceedings of the Workshop on Searching Spontaneous Conversational Speech, ACM Multimedia 2009* Beijing, China, 2009.

[33] G. Salton, *Automatic Information Organisation and Retrieval.* New York: McGraw-Hill, 1968.

[34] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*: ACM Press / Addison-Wesley, 1999.

[35] D. A. Grossman and O. Frieder, *Information Retrieval: Algorithms and Heuristics*: Kluwer Academic Publishers, 1998.

[36] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 2nd edition ed. New Jersey: Prentice Hall, 2008.

[37] D. J. Harper, "Relevance Feedback in Document Retrieval Systems: An Evaluation of Probabilistic Strategies." vol. Ph.D. thesis Cambridge: Cambridge University, 1980.

[38] E. A. Fox, S. Betrabet, M. Koushik, and W. Lee, "Extended Boolean models," *Information Retrieval: Data Structures and Algorithms,* pp. 393-418, 1992.

[39] V. N. GUDIVADA, V. V. RAGHAVAN, W. I. GROSKY, and R. KASANAGOTTU, "Information retrieval on the World Wide Web," *IEEE internet computing,* vol. 1, pp. 58-68, 1997.

[40] S. K. M. Wong, W. Ziarko, V. V. Raghavan, and P. C. N. Wong, "Extended Boolean query processing in the generalized vector space model," *Information Systems,* vol. 14, pp. 47-63, 1989.

[41] W. S. Cooper, "Getting beyond Boole," *Information Processing and Management,* vol. 24, pp. 243-248, 1988.

[42] S. C. Cater and D. H. Kraft, "TIRS: a topological information retrieval system satisfying the requirements of the Waller-Kraft wish list," in *Proceedings of the10th annual international ACM SIGIR conference on Research and development in information retrieval* New Orleans, Louisiana, United States: ACM, 1987, pp. 171-180.

[43] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*: McGraw-Hill, Inc., 1986.

[44] C. D. Paice, "Soft evaluation of Boolean search queries in information retrieval systems," *Information Technology Research Development Applications,* vol. 3, pp. 33-41, 1984.

[45] A. Bookstein, "Probability and fuzzy-set application to information retrieval," *Annual Review of Information Science and Technology,* vol. 20, pp. 117-151, 1985.

[46] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Comunication of the ACM,* vol. 18, pp. 613-620, 1975.

[47] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*: Addison-Wesley Longman Publishing Co., Inc., 1989.

[48] G. Salton and M. E. Lesk, "Computer Evaluation of Indexing and Text Processing," *Journal of the ACM* vol. 15, pp. 8-36, 1968.

[49] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management,* vol. 24, pp. 513-523, 1988.

[50] A. Singhal, C. Buckley, M. Mitra, and G. Salton, "Pivoted Document Length Normalization," Cornell University 1995.

[51] S. E. Robertson, S. Walker, and M. M. Beaulieu, "Okapi at TREC-7: automatic ad hoc, filtering, VCL and interactive track," in *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*: National Institute of Standards and Technology 1999.

[52] G. Salton, "Automatic Text Analysis," *Science* vol. 168, pp. 335 - 343, 1970.

[53] G. Salton, "A Comparison Between Manual and Automatic Indexing Methods," *American Documentation,* vol. 20, 1969.
[54] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval.* New York, NY: McGraw-Hill, 1983.
[55] S. Ghahramani, *Fundamentals of Probability*: Prentice Hall, 2000.
[56] T. T. Soong, *Fundamentals of Probability and Statistics for Engineers*: Wiley-Interscience 2004.
[57] D. C. Blair and M. E. Maron, "An evaluation of retrieval effectiveness for a full-text document-retrieval system," *Communications of the ACM* vol. 28, pp. 289-299, 1985.
[58] M. E. Maron and J. L. Kuhns, "On Relevance, Probabilistic Indexing and Information Retrieval," *Journal of the ACM* vol. 7, pp. 216-244, 1960.
[59] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science,* vol. 27, pp. 129-146, 1976.
[60] S. E. Robertson, S. Walker, S. Jones, M. M. Beaulieu, and M. Gatford, "Okapi at TREC 3," in *Proceedings of the third Text Retrieval conference* Gaithersburg, USA, 1994.
[61] K. S. Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments (parts 1 and 2)," *Information Processing and Management,* vol. 36, pp. 779-840, 2000.
[62] G. Amati and C. J. V. Rijsbergen, *Probabilistic models of information retrieval based on measuring the divergence from randomness* vol. 20. New York: ACM, 2002.
[63] B. He and I. Ounis, "A study of parameter tuning for term frequency normalization," in *Proceedings of the twelfth international conference on Information and knowledge management* New Orleans, LA, USA: ACM, 2003.
[64] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson, "Terrier Information Retrieval Platform " *Lecture notes in computer science: Advances in Information Retrieval,* vol. 3408/2005, pp. 517-519, 2005.
[65] J. M. Ponte, "A language modeling approach to information retrieval," in *Computer Science.* vol. Ph.D. Thesis: University of Massachusetts Amherst, 1998.
[66] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* Melbourne, Australia: ACM, 1998.
[67] A. Bookstein, "Fuzzy requests: An approach to weighted boolean searches," *Journal of the American Society for Information Science and Technology,* vol. 31, pp. 240 - 247, 1979.
[68] B. P. BUCKLES and F. E. PETRY, "A FUZZY REPRESENTATION OF DATA FOR RELATIONAL DATABASES," *Fuzzy Sets and Systems,* vol. 7, pp. 213-226, 1982.
[69] H. Turtle and W. B. Croft, "Inference networks for document retrieval," in *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval* Brussels, Belgium: ACM, 1990.

[70] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis " *Journal of the American Society for Information Science* vol. 41, pp. 391–407, 1990.

[71] T. Landauer, P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes,* vol. 25, pp. 259-284, 1998.

[72] S. T. Dumais, "Latent semantic indexing (LSI): TREC-3 report " in *Proceedings of the third Text retrieval Conference (TREC-3)*, 1995, pp. 219-230.

[73] M. Gordon, "Probabilistic and genetic algorithms in document retrieval," *Communications of the ACM* vol. 31, pp. 1208-1218, 1988.

[74] J.-J. Yang and R. R. Korfhage, "Query modification using genetic algorithms in vector space models," *International Journal of Expert Systems* vol. 7, pp. 165-191, 1994.

[75] D. H. Kraft, F. E. Petry, B. P. Buckles, and T. Sadashan, "The use of genetic programming to build queries for information retrieval," *IEEE Symposium on Evolutionary Computation,* vol. 1, pp. 468-473 1994.

[76] R. Wilkinson and P. Hingston, "Incorporating the vector space model in a neural network used for document retrieval," *Library Hi Tech,* vol. 10, pp. 69-75, 1992.

[77] K. L. Kwok, "A neural network for probabilistic information retrieval," in *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval* Cambridge, Massachusetts, United States: ACM, 1989.

[78] F. Crestani, "Implementation and evaluation of a relevance feedback device based on neural networks " *From Natural to Artificial neural Computation: International Workshop on Artificial Neural Networks,* vol. 930, pp. 597-604, 1995.

[79] J. J. Rocchio, "Relevance feedback in information retrieval," *The SMART Retrieval System: Experiments in Automatic Document Processing,* pp. 313-323, 1971.

[80] J. J. Rocchio, "Document retrieval systems; optimization and evaluation.." vol. Ph.D. Thesis: Harvard University, 1966.

[81] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science,* vol. 41, pp. 288-297, 1990.

[82] C. v. Rijsbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," *Journal of Documentation,* vol. 33, pp. 106-119, 1977.

[83] D. J. Harper and C. J. V. RIJSBERGEN, "An Evaluation of Feedback in Document Retrieval Using Co-Occurrence Data," *Journal of Documentation,* vol. 34, pp. 189 - 216, 1978.

[84] W. B. Croft and D. J. Harper, "Using probabilistic models of document retrieval without relevance information," *Journal of Documentation,* vol. 35, pp. 285 - 295, 1979.

[85] K. S. Jones, "Search term relevance weighting given little relevance information," *Journal of Documentation,* vol. 35, pp. 30 - 48, 1979

[86] A. F. Smeaton and C. J. v. Rijsbergen, "The Retrieval Effects of Query Expansion on a Feedback Document Retrieval System " *Computer Journal,* vol. 26, pp. 239-246 1983.

[87] D. Harman, "Relevance feedback revisited," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* Copenhagen, Denmark: ACM, 1992.

[88] C. Buckley, A. Singhal, and M. Mitra, "New Retrieval Approaches Using SMART: TREC 4," *TREC 4,* pp. 25-48, 1995.

[89] J. Xu and W. B. Croft, "Query expansion using local and global document analysis," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* Zurich, Switzerland: ACM, 1996.

[90] G. Amati, "Probabilistic Models for Information Retrieval based on Divergence from Randomness," in *Department of Computing Science,*. vol. Ph.D. thesis Glasgow: University of Glasgow, 2003.

[91] E. M. Voorhees, "Using WordNet to disambiguate word senses for text retrieval," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* Pittsburgh, Pennsylvania, United States: ACM, 1993.

[92] E. M. Voorhees, "Query expansion using lexical-semantic relations," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* Dublin, Ireland: Springer-Verlag New York, Inc., 1994.

[93] K. S. Jones and E. O. Barber, "What Makes An Automatic Keyword Classification Effective?," *Journal of the American Society for Information Science,* vol. 22, pp. 166-175, 1971.

[94] H. Schutze and J. O. Pedersen, "A cooccurrence-based thesaurus and two applications to information retrieval," *Information Processing & Management,* vol. 33, pp. 307-318, 1997.

[95] Y. Jing and W. B. Croft, "An Association Thesaurus for Information Retrieval," in *Proceedings of the RIAO 94 Conference Proceedings*, 1994.

[96] Y. Qiu and H.-P. Frei, "Concept based query expansion," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* Pittsburgh, Pennsylvania, United States: ACM, 1993.

[97] H. Kucera, W. N. Francis, J. B. Carroll, and W. F. Twaddell, *Computational Analysis of Present Day American English* Providence: Brown University, 1967.

[98] C. Fox, "A stop list for general text," *ACM SIGIR Forum* vol. 24, pp. 19-21, 1990.

[99] W. J. Wilbur and K. Sirotkin, "The automatic identification of stop words," *Journal of Information Science,* vol. 18, pp. 45-55, 1992.

[100] R. T. Lo, B. He, and I. Ounis, "Automatically Building a Stopword List for an Information Retrieval System," *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR 2005),* pp. 3–8, 2005.

[101] S.-B. Kim, H.-C. Seo, and H.-C. Rim, "Information retrieval using word senses: root sense tagging approach," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* Sheffield, United Kingdom: ACM, 2004.

[102] S. Liu, C. Yu, and W. Meng, "Word sense disambiguation in queries," in *Proceedings of the 14th ACM international conference on Information and knowledge management* Bremen, Germany: ACM, 2005.

[103] C. Stokoe, M. P. Oakes, and J. Tait, "Word sense disambiguation in information retrieval revisited," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* Toronto, Canada: ACM, 2003.

[104] W. B. Frakes, "Stemming Algorithms," *Information Retrieval: Data Structures and Algorithms,* 1992.

[105] R. Krovetz, "Viewing morphology as an inference process," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* Pittsburgh, Pennsylvania, United States: ACM, 1993.

[106] J. B. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics,* vol. 11, pp. 22-31, 1968.

[107] M. F. Porter, "An algorithm for suffix stripping," *Program,* vol. 14, pp. 130-137, 1980.

[108] J. A. Shaw and E. A. Fox, "Combination of Multiple Searches," *Proceedings of the third Text REtrieval Conference (TREC-3),* pp. 105-108, 1994.

[109] J. H. Lee, "Combining multiple evidence from different properties of weighting schemes," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* Seattle, Washington, United States: ACM, 1995.

[110] E. A. Fox, M. P. Koushik, J. Shaw, R. Modlin, and D. Rao, "Combining Evidence from Multiple Searches " in *Proceedings of the first Text REtrieval Conference (TREC-1)* Gaithersburg, Maryland, USA: National Institute of Standards and Technology (NIST) 1992.

[111] J. H. Lee, "Analyses of multiple evidence combination," in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* Philadelphia, Pennsylvania, United States: ACM, 1997.

[112] C. C. Vogt and G. W. Cottrell, "Predicting the performance of linearly combined IR systems," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* Melbourne, Australia: ACM, 1998.

[113] B. Ramabhadran, O. Siohan, and G. Zweig, "Use of Metadata to Improve Recognition of Spontaneous Speech and Named Entities," *Proceedings of the INTERSPEECH 2004 - ICSLP,* pp. 381-384, 2004.

[114] R. M. Terol, M. Palomar, P. Martinez-Barco, F. Llopis, R. Mu˜noz, and E. Noguera, "The University of Alicante at CL-SR Track," *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 769–772, 2006.

[115] A. M. Lam-Adesina and G. J. F. Jones, "Dublin City University at CLEF 2005: Cross-Language Speech Retrieval (CL-SR) Experiments," *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 792 – 799, 2006.

[116] J. Wang and D. W. Oard, "CLEF-2005 CL-SR at Maryland:Document and Query Expansion Using Side Collections and Thesauri," *Lecture notes in computer sci-*

ence: Accessing Multilingual Information Repositories, vol. 4022/2006, pp. 800–809, 2006.

[117] F. L´opez-Ostenero, V. ı. Peinado, V. ı. Sama, and F. Verdejo, "UNED@CL-SR CLEF 2005: Mixing Different Strategies to Retrieve Automatic Speech Transcriptions," *Lecture notes in computer science: Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 783–791, 2006.

[118] D. He and J.-W. Ahn, "Pitt at CLEF05: Data Fusion for Spoken Document Retrieval," *Accessing Multilingual Information Repositories,* vol. 4022/2006, pp. 773–782, 2006.

[119] C. L. A. Clarke, "Waterloo Experiments for the CLEF05 SDR Track," *Accessing Multilingual Information Repositories,* vol. 4022/2006, 2006.

[120] R. M. Terol, P. Martinez-Barco, and M. Palomar, "Applying Logic Forms and Statistical Methods to CL-SR Performance," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 766–769, 2007.

[121] G. J. F. Jones, K. Zhang, and A. M. Lam-Adesina, "Dublin City University at CLEF 2006: Cross-Language Speech Retrieval (CL-SR) Experiments," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 794–802, 2007.

[122] G. J. F. Jones, K. Zhang, E. Newman, and A. M. Lam-Adesina, "Examining the Contributions of Automatic Speech Transcriptions and Metadata Sources for Searching Spontaneous Conversational Speech," in *SIGIR 2007 workshop: Search in Spontaneous Conversational Speech workshop* Amsterdam, 2007.

[123] S. Robertson, H. Zaragoza, and M. Taylor, "Simple BM25 extension to multiple weighted fields," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management* Washington, D.C., USA: ACM, 2004.

[124] J. Wang and D. W. Oard, "CLEF-2006 CL-SR at Maryland: English and Czech," *Lecture notes in computer science: Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 786–793, 2007.

[125] R. Aly, D. Hiemstra, R. Ordelman, L. v. d. Werff, and F. d. Jong, "XML Information Retrieval from Spoken Word Archives," *Evaluation of Multilingual and Multi-modal Information Retrieval,* vol. 4730/2007, pp. 770–777, 2007.

[126] M. Lease and E. Charniak, "A Dirichlet-Smoothed Bigram Model for Retrieving Spontaneous Speech," *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* pp. 687–694, 2008.

[127] Y. Zhang, G. J. F. Jones, and K. Zhang, "Dublin City University at CLEF 2007: Cross-Language Speech Retrieval Experiments," *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* vol. 5152, pp. 703–711, 2008.

[128] B. Huurnink, "The University of Amsterdam at the CLEF Cross Language Speech Retrieval Track 2007," *Lecture Notes in Computer Science: Advances in Multi-*

lingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers, vol. 5152, 2008.

[129] G.-A. Levow, "University of Chicago at the CLEF 2007 Cross-language Speech Retrieval Track," *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* vol. 5152, 2008.

[130] J. P. Callan, W. B. Croft, and S. M. Harding, "The INQUERY retrieval system," in *Proceedings of the third International Conference on Database and Expert Systems Applications* Valencia, Spain: Springer-Verlag, 1992.

[131] M. C. Díaz-Galiano, M. T. Martín-Valdivia, M. A. García-Cumbreras, and L. A. Ureña-López, "Using Information Gain to Filter Information in CLEF CL-SR Track " *Lecture Notes in Computer Science: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers,* vol. 5152, pp. 719-724, 2008.

[132] A. Singhal, J. Choi, D. Hindle, and F. Pereira, "AT&T at TREC-6: SDR Track," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[133] M. A. Siegler, M. J. Witbrock, S. T. Slattery, K. Seymore, R. E. Jones, and A. G. Hauptmann, "Experiments in Spoken Document Retrieval at CMU," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[134] S. Walker, S. E. Robertson, M. Boughanem, G. J. F. Jones, and K. S. Jones, "Okapi at TREC-6 Automatic ad hoc, VLC, routing, filtering and QSDR," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[135] A. F. Smeaton, F. Kelledy, and G. Quinn, "Ad hoc Retrieval Using Thresholds, WSTs for French Monolingual Retrieval, Document-at-a-Glance for High Precision and Triphone Windows for Spoken Documents," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[136] B. Mateev, E. Munteanu, P. Sheridan, M. Wechsler, and P. Schauble, "ETH TREC-6: Routing, Chinese, Cross-Language and Spoken Document Retrieval," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[137] F. Crestani, M. Sanderson, M. Theophylactou, and M. Lalmas, "Short Queries, Natural Language and Spoken Document Retrieval: Experiments at Glasgow University," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[138] M. Fuller, M. Kaszkiel, C. L. Ng, P. Vines, R. Wilkinson, and J. Zobel, "MDS TREC6 Report," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[139] P. Schone, J. L. Townsend, T. H. Crystal, and C. Olano, "Text Retrieval via Semantic Forests," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[140] P. Schone and D. Nelson, "A Dictionary-Based Method for Determining Topics in Text and Transcribed speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech, & Signals Processing* Atlanta, Georgia, 1996.

[141] D. Abberley, S. Renals, G. Cook, and T. Robinson, "The THISL Spoken Document Retrieval System," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[142] D. W. Oard and P. Hackett, "Document Translation for Cross-Language Text Retrieval at the University of Maryland," in *Proceedings of the Sixth Text REtrieval Conference (TREC-6)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1997.

[143] M. A. Siegler, A. Berger, M. J. Witbrock, and A. G. Hauptmann, "Experiments in Spoken Document Retrieval at CMU," in *Proceedings of the Seventh Text REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[144] S. E. Johnson, P. Jourlin, G. L. Moore, K. S. Jones, and P. C. Woodland, "Spoken Document Retrieval For TREC-7 At Cambridge University," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[145] M. Fuller, M. Kaszkiel, D. Kim, C. Ng, J. Robertson, R. Wilkinson, M. Wu, and J. Zobel, "TREC 7 Ad Hoc, Speech, and Interactive tracks at MDS/CSIRO," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[146] G. D. Henderson, P. Schone, and T. H. Crystal, "Text Retrieval via Semantic Forests: TREC7," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[147] P. Nowell, "Experiments in Spoken Document Retrieval at DERA-SRU," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[148] R. Ekkelenkamp, W. Kraaij, and D. v. Leeuwen, "TNO TREC7 site report: SDR and filtering," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[149] D. W. Oard, "TREC-7 Experiments at the University of Maryland," in *Proceedings of the SeventhText REtrieval Conference (TREC-7)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1998.

[150] A. Singhal and F. Pereira, "Document expansion for speech retrieval," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* Berkeley, California, United States: ACM, 1999.

[151] M. Fuller, M. Kaszkiel, S. Kimberley, C. Ng, R. Wilkinson, M. Wu, and J. Zobel, "The RMIT/CSIRO Ad Hoc, Q&A, Web, Interactive, and Speech Experiments at TREC 8," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[152] M. Franz, J. S. McCarley, and R. T. Ward, "Adhoc, Cross-language and Spoken Document Information Retrieval at IBM," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[153] S. E. Johnson, P. Jourlin, K. S. Jones, and P. C. Woodland, "CMU Spoken Document Retrieval in Trec-8: Analysis of the role of Term Frequency TF," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[154] W. Kraaij, R. Pohlmann, and D. Hiemstra, "Twenty-One at TREC-8: using Language Technology for Information Retrieval," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* Gaithersburg, Maryland: National Institute of Standards and Technology (NIST), 1999.

[155] S. Banerjee and T. Pedersen, "The Design, Implementation and Use of the Ngram Statistics Package " in *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics* Mexico City, Mexico, 2003.

[156] C. Buckley, G. Salton, and J. Allan, "Automatic Retrieval With Locality Information Using SMART," *Proceedings of theText REtrieval Conferenc (TREC-1),* pp. 59-72, 1992.

[157] M. Alzghool and D. Inkpen, "Model Fusion Experiments for the Cross Language Speech Retrieval Task at CLEF 2007," in *Working Notes of the CLEF- 2007 Evaluation* Budapest-Hungary, 2007.

[158] P. Pecina, P. Hoffmannov´a, G. J. F. Jones, Y. Zhang, and D. W. Oard, "Overview of the CLEF-2007 Cross Language Speech Retrieval Track," in *Working Notes of the CLEF- 2007 Evaluation* Budapest-Hungary: CLEF2007, 2007.

[159] M. Montague, "Metasearch: Data Fusion for Document Retrieval," in *Computer science*. vol. Ph.D. Hanover: Dartmouth College, 2002.

[160] D. A. Hull, J. O. Pedersen, and H. Schütze, "Method combination for document filtering," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* Zurich, Switzerland: ACM, 1996.

[161] P. S. m. d. Laplace, *Essai philosophique sur les probabilites*. Paris: Mme. Ve. Courcier, 1814.

[162] G. J. Lidstone, "Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities," *Transactions of the Faculty of Actuaries,* vol. 8, pp. 182-192, 1920.

[163] D. He and D. Wu, "Toward a Robust Data Fusion for Document Retrieval," in *Proceedings of the 2008 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'08)* Beijing, China, 2008.