國立臺灣大學電機資訊學院電信工程學研究所

博士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

語音數位內容檢索 — 相關回饋、圖論及語意

Spoken Content Retrieval – Relevance Feedback, Graphs and Semantics

李宏毅

Lee Hung-Yi

指導教授：李琳山 教授

Advisor: Lee Lin-Shan, Ph.D.

中華民國一零一年七月

July, 2012

摘要

一般的語音資訊檢索可以分成兩個階段。語音辨識引擎先將語料庫中的語音資訊轉寫成文字並儲存起來；然後在檢索時，就直接把文字資訊檢索的方法套用在這些辨識結果上。如果語音辨識引擎可以正確的將語音轉寫成文字，上述架構當然可以得到良好的結果，然而在語音辨識系統正確率較差的情況下，這樣的架構勢必會造成語音資訊檢索的效能大幅下降。本論文的核心思想就是要突破上述架構中語音資訊檢索因完全仰賴語音辨識結果所造成的效能限制，這將會是語音資訊檢索這個領域未來非常重要的發展方向。

本論文首先提出了以使用者相關回饋來重估測辨識系統的聲學模型參數的新技術。有別於傳統的聲學模型訓練法，本論文以提升檢索效能做為聲學模型訓練的目標，並將檢索系統以排序結果進行評估的特性在聲學模型訓練的過程中加以考量。另一方面，本論文提出了以聲學特徵參數做為機器學習特徵的想法，這個方法成功的被實作在虛擬回饋的架構下。其次，為了彌補在辨識過程中所漏失的資訊，本論文提出以聲學特徵相似度來改進語音資訊檢索的想法，這個想法可以被用在虛擬回饋以及圖學基礎之重排序上。最後，雖然今日語音檢索的研究仍集中在提升口述語彙偵測的效能，但本論文進一步考慮了語意檢索，目標在找出語意相關的語音文件，而不僅僅是找出包含查詢詞的文件。本文提出了以聲學特徵相似度來提升詞頻估測準確率的方法，這個方法可以進一步提升語意檢索中的語言檢索模型、文件擴展、查詢詞擴展等技術之效能。

Abstract

Multimedia content over the Internet is very attractive, while the spoken part of such content very often tells the core information. Therefore, spoken content retrieval will be very important in helping users retrieve and browse efficiently across the huge qualities of multimedia content in the future. There are usually two stages in typical spoken content retrieval approaches. In the first stage, the audio content is recognized into text symbols by an Automatic Speech Recognition (ASR) system based on a set of acoustic models and language models. In the second stage, after the user enters a query, the retrieval engine searches through the recognition output and returns to the user a list of relevant spoken documents or segments. If the spoken content can be transcribed into text with very high accuracy, the problem is naturally reduced to text information retrieval. However, the inevitable high recognition error rates for spontaneous speech under a wide variety of acoustic conditions and linguistic context make this never possible. In this thesis, the above standard two-stage architecture is completely broken, and the two stages of recognition and retrieval are mixed up and considered as a whole. A set of approaches beyond retrieving over recognition output has been developed here. This idea is very helpful for spoken content retrieval, and may become one of the main future directions in this area.

To consider the two stages of recognition and retrieval as a whole, it is proposed to adjust the acoustic model parameters borrowing the techniques of discriminative training but based on user relevance feedback. The problem of retrieval oriented acoustic model re-estimation is different from the conventional acoustic model training approaches for speech recognition in at least two ways:

1. The model training information includes only whether a spoken segment is relevant to a query or not; it does not include the transcription of any utterance.

2. The goal is to improve retrieval performance rather than recognition accuracy.

A set of objective functions for retrieval oriented acoustic model re-estimation is proposed to take the properties of retrieval into consideration.

There have been some previous works in spoken content retrieval taking advantage of the discriminative capability of machine learning methods. Different from the previous works that derive information from recognition output as features, acoustic vectors such as MFCC are taken as the features for discriminating relevant and irrelevant segments, and they are successfully applied on the scenario of Pseudo Relevance Feedback (PRF).

The recognition process can be considered as "quantization", in which the acoustic vector sequences are quantized into word symbols. Because different vector sequences may be quantized into the same symbol, much of the information in the spoken content may be lost in the stage of speech recognition. Information directly from the acoustic vector space is considered to compensate for the recognition output in this thesis. This is realized by either PRF or a graph-based re-ranking approach considering the similarity structure among all the segments retrieved. This approach is successfully applied on not only word-based retrieval system but also subword-based system, and these approaches improve the results of Out-of-Vocabulary (OOV) queries as well.

The task of Spoken Term Detection (STD) is mainly considered in this thesis, for which the goal is simply returning spoken segments that contain the query terms. Although most works in spoken content retrieval nowadays continue to focus on STD, in this thesis a more general task is also considered: to retrieve the spoken documents se-

mantically related to the queries, no matter the query terms are included in the spoken documents or not. Taking ASR transcriptions as text, the techniques such as latent semantic analysis or query expansion developed for text-based information retrieval can be directly applied for this task. However, the inevitable recognition errors in ASR transcriptions degrade the performance of these techniques. To have more robust semantic retrieval of spoken documents, the expected term frequencies derived from the lattices are enhanced by acoustic similarity with a graph-based approach. The enhanced term frequencies improve the performance of language modelling retrieval approach, document expansion techniques based on latent semantic analysis, and query expansion methods considering both words and latent topic information.

Contents

List of Figures

List of Tables

Part I

Introduction and Background Review

Chapter 1 Introduction

1.1 Spoken Content Retrieval

Ever increasing computing power and connectivity bandwidth together with falling storage costs result in an overwhelming amount of data of various types being produced, exchanged and stored. Consequently, retrieval has emerged as a key application as more and more data are being saved. Spoken content retrieval did not receive much attention in the past due to the fact that large collections of spoken material were not available because of storage constraints. However, as web sites providing multimedia are becoming more and more popular, this research topic has attracted lots of attentions now. Since the subjects, topics, and core concepts of such multimedia content can very often be identified based on the speech information within the audio part of the content, spoken content retrieval will be very important in helping users retrieve and browse efficiently across the huge qualities of multimedia content in the future [1]. The basic scenario for spoken content retrieval is that the user enters a *query* in text form into the retrieval system, and the system returns a list of rank-ordered *spoken documents* or *spoken segments*. Spoken segments are the portions of longer spoken documents, which can be as short as individual spoken utterances. Without specially mentioned, we assume the queries for spoken content retrieval are in text form in this thesis. Although spoken queries may be used instead of text in some applications, it is out of the scope of this thesis.

An intuitive but widely applied approach for spoken content retrieval is "use an automatic speech recognition (ASR) system to transcribe the spoken content first, and then apply state-of-the-art text-based retrieval approaches on the transcriptions". Therefore,

there are usually two stages in typical spoken content retrieval approaches [2]. In the first stage, the audio content is recognized into word sequences by ASR system based on a set of acoustic models and language models. In the second stage, after the user enters a query, the retrieval engine searches through the recognition output and returns to the user a list of relevant spoken documents or segments. This strategy works reasonably well when the speech recognition output is mostly correct.

Actually, the above approach made a success in Text REtrieval Conference (TREC) Spoken Document Retrieval (SDR) track and achieved very similar accuracy performances compared with human transcripts. Therefore, people considered the SDR problem as a "solved" problem at the time [3]. However, the SDR track for TREC was conducted on broadcast news with relatively low recognition error rates. When human extended spoken content retrieval to more challenging task such as telephone conversations, meetings, or lecture courses whose word error rates are sometimes higher than 50%, the above strategy was not sufficient to build a retrieval system with reasonable performance.

Decreasing the word error rates of the ASR modules inherent in the spoken content retrieval systems certainly improves the retrieval performance. Whenever the ASR module is able to correctly transcribe any audio content into its corresponding word strings, spoken content retrieval would be reduced to text-based information retrieval. However, it is hard to imagine that ASR system can be perfect in the near future. Although lots of efforts have been put into the research for ASR, it is still very challenging for the state-of-the-art ASR systems to perfectly recognize audio content, in particular those from the Internet with widely variant acoustic and linguistic conditions.

A widely considered approach is indexing the multiple hypotheses of spoken con-

tent. It has been proved to be helpful for dealing spoken archive with high recognition errors. It has been shown that retrieval based on multiple hypotheses offered over 30% improvements of Maximum F-measure over one-best transcription for a term detection task in broadcast news, switchboard and teleconferences [4]. However, for poor recognition accuracies, even if the correct word hypotheses can be included in the lattice, many incorrect noisy hypotheses disturb the results. Therefore, the performance of spoken content retrieval is still inevitably dominated by ASR performance based on the lattices.

How about break through the typical two-stage architecture in spoken content retrieval? Since the input query is in text form, and the target spoken archive is audio files, to bridge the gap between text and audio, the recognition transforming speech into text may be indispensable. However, because the recognition output can only be regarded as a very rough approximation for the spoken content especially when recognition is poor, the retrieval techniques solely based on the recognition output are not able to take the advantage of the possibly useful information lost during the recognition. Therefore, it is certainly beneficial to regard the spoken content as "spoken content" itself in the task of spoken content retrieval. In this thesis, a set of novel retrieval approaches beyond recognition output are proposed following the above philosophy, which consider the characteristics of the spoken content in the retrieval process. These methods are believed to surmount the limitation from the imperfect ASR system to some extent.

Two subtasks of spoken content retrieval are considered in this thesis. The first one is *Spoken Term Detection*, for which the goal is simply returning spoken segments that contain the query terms. The second one is *Semantic Retrieval of Spoken Content*, in which the retrieval system retrieves the spoken documents semantically related to the

queries (no matter the query terms are included in the spoken documents or not).

1.1.1 Spoken Term Detection

The recognition and retrieval system should be considered as a whole. Since the retrieval process is mostly based on the recognition output like transcriptions or lattices, the parameters of the recognition models may influence the performance of the whole retrieval system. Hence, better acoustic models conceivably improve the retrieval performance. To obtain better acoustic models, conventionally a set of utterances with their manual transcriptions are needed, but here the better acoustic model parameters are estimated based on user relevance feedback. The relevance feedback oriented acoustic model estimation aims at improving the retrieval performance instead of recognition accuracies, and a set of objective functions for relevance feedback is proposed, which take the properties for retrieval into consideration.

It has been known for long that the retrieval task can be considered as binary classification, and the binary classifiers trained from the data obtained by relevance feedback via the machine learning methods can be used for identifying the relevance of an object with respect to a specific query. The same approach can be naturally applied on spoken content retrieval. However, it is unclear which kinds of information in the spoken content should be represented as features for training the models accurately discriminating the relevant objects from the irrelevant ones. The features directly derived from the acoustic vector sequences (such as Mel-Frequency Cepstral Coefficients (MFCC)) of the spoken content is found to be very helpful in this thesis. This approach is successfully applied on pseudo-relevance feedback, and thus the system performance is improved automatically

without involving any user.

Since the acoustic vector sequences representing different occurrences of the same term should be similar in some way, while very different vector sequences very possibly imply different terms, if some audio examples for the desired query are available, it is possible to judge the correctness of the retrieved spoken objects based on the similarity of the acoustic vector sequences of these objects to those of the given examples. Those examples can be obtained by pseudo-relevance feedback. This idea can be moved one step forward via considering the relevance of the spoken content in a more global way with a graph. Since the acoustic similarity is completely independent to the recognition output, the methods developed based on acoustic similarities may be robust to the recognition errors. These example-based approaches are verified not only useful for IV queries but also OOV queries. Moreover, it is also successfully applied on the spoken archive with many different speakers.

1.1.2 Semantic Retrieval of Spoken Content

Most of researches on spoken content retrieval nowadays focus on spoken term detection, but this is insufficient because users naturally prefer that the technologies can return all the objects that the user really wants, regardless of whether the query terms are contained or not, so semantic retrieval of spoken content is also considered in this thesis. The *Concept Matching* techniques [1] which have been widely studied in text-based information retrieval are desired in this task. Taking ASR transcriptions as pure text, the concept matching techniques developed for text-based information retrieval can be directly applied on semantic retrieval of spoken content. However, since these techniques

were developed for text without errors, the inevitable recognition errors in ASR transcriptions definitely degrade the performance, even though the retrieval is based on the lattices. To tackle this problem, instead of simply estimating the term distributions in the spoken documents from the lattices, they are refined based on the concept that similar terms may exhibit similar acoustic vector sequences. This method is able to enhance the techniques developed for text-based retrieval including query expansion and document expansion.

1.2 Organization of the Thesis

Part I of this thesis consisting of Chapter 1 to 3 advocates for the introduction and review of background knowledge. In Chapter 2, general issues related to spoken content retrieval are briefly introduced, and an overview for the scenarios of relevance feedback is given in Chapter 3. Part II of this thesis introduces the new techniques proposed. Chapter 4 to 6 are for spoken term detection, in which the objects to be retrieved are spoken segments, and a spoken segment is taken as relevant if it includes the query term. In chapter 4, a set of relevance feedback oriented acoustic model re-estimation methods are proposed. In chapter 5, SVM models are trained for identifying the segments' relevance from acoustic vectors in the scenario of pseudo-relevance feedback. In Chapter 6, a set of example-based methods directly employing the acoustic similarities between the spoken segments are proposed. The relevance of a spoken segment can be either judged by some audio examples from pseudo-relevance feedback, or determined by the similarity structure for the retrieved segments with a graph. The Chapter 7 of this thesis considers a more general task for spoken content retrieval, in which the retrieval system retrieves the spoken documents semantically related to the queries, and the term distributions in the spoken

documents are refined based on the concept of acoustic vector similarity to enhance the text-based techniques. Finally, the conclusion and future work are given in Chapter 8.

Chapter 2 Spoken Content Retrieval

In this chapter, a brief introduction about spoken content retrieval is given. For the interested readers, please refer to the references [1,2,5,6].

2.1 Basic Idea

The basic idea of information retrieval is after a user enters a query Q, the retrieval system returns a list of objects x ranked according to their relevance with respect to the query. The definition for the relevance of an object is task-dependent or even user-dependent [1]. The relevance of each object x with respect to a query Q is evaluated by a relevance score function $S(Q, x)$ in the retrieval system. This relevance score function $S(Q, x)$ can be learned from a set of training data, or designed by the system designers heuristically. After a query Q is requested from the user, the system computes $S(Q, x)$ for each object x in the database and returns a list of objects sorting by $S(Q, x)$ to the user. For spoken content retrieval discussed in this thesis, the query Q is a string of words, and the objects x to be returned are spoken documents or segments.

Most of the following discussions in this chapter are for Spoken Term Detection (STD), in which spoken segments are objects to be retrieved, and a spoken segment x is relevant if it includes the query term. These discussions may be generalized to other tasks in spoken content retrieval as well. The STD task is trivial for one-best transcription. Since the user wants to find the spoken segments containing the query term, the STD

[1] Relevance can be roughly understood as the preference for the user.

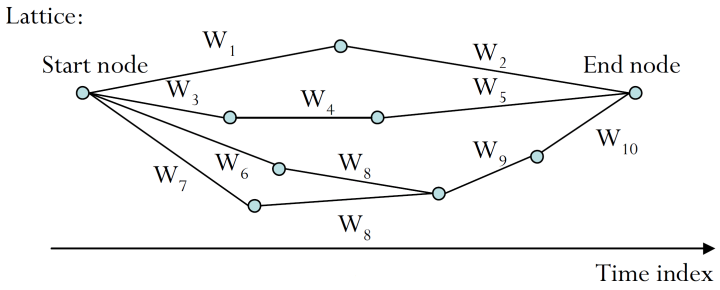
system can simply search through the spoken segments in the data collection, and return the spoken segments which contain the query term in their transcriptions. However, the unavoidable recognition errors usually lead to poor performance for the above approach. A more robust approach is computing the expected query term frequency $E[Q|x]$ for each spoken segment x. The expected term frequency $E[Q|x]$ or its variants are widely used in the STD task as the relevance score function $S(Q, x)$ [7–12]. In other words, after a query term Q is entered, the system ranks the segments x according to $E[Q|x]$. Theoretically, $E[Q|x]$ is defined as

$$E[Q|x] = \sum_{all\ possible\ word\ sequences\ u} N(u, Q)P(u|x), \tag{2.1}$$

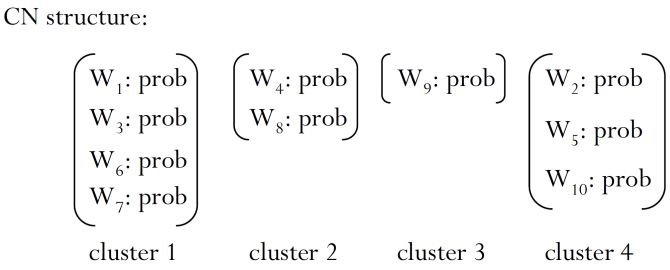
where $N(u, Q)$ is the occurrence count of query Q in u, and $P(u|x)$ is the posterior probability of a word sequence u given the spoken segment x based on a set of acoustic and language models. It seems impossible to compute the posterior probabilities for all possible word sequences, but (2.1) can be approximated by only considering the word sequences u included in the lattice structure introduced in the next subsection.

2.2 Lattices

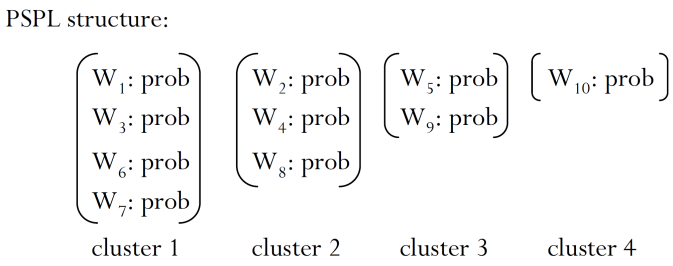
Given an utterance, the ASR system can not only return the word sequence with the largest posterior probability as the transcription, but also return all of the word sequences whose probabilities are larger than a threshold based on a set of intrinsic acoustic and language models. Those word sequences are usually merged into a lattice structure as Fig 2.1a, in which four word sequences, $\{W_1, W_2\}$, $\{W_3, W_4, W_5\}$, $\{W_6, W_8, W_9, W_{10}\}$ and $\{W_7, W_8, W_9, W_{10}\}$, are embedded in the lattice. Even though the transcription is not



(a) Lattice



(b) Confusion Network (CN)



(c) Position Specific Posterior Lattice (PSPL)

Figure 2.1: Lattice and sausage-like structures.

correct, it is often possible to find the correct word sequence in the lattice.

Suppose each spoken segment x is first transcribed into a lattice $W(x)$, then (2.1) can be approximated by only considering the word sequences in the lattice as

$$E[Q|x] = \sum_{u \in W(x)} N(u, Q) P(u|x), \tag{2.2}$$

where u is an allowed word sequence in the lattice $W(x)$, $N(u, Q)$ is the occurrence count of query Q in u, and

$$P(u|x) = \sum_{u \in W(x)} N(u, Q) \frac{P(x|u)P(u)}{\sum_{u \in W(x)} P(x|u)P(u)}, \tag{2.3}$$

where $P(x|u)$ is the likelihood for observation sequence of segment x given the word sequence u based on the acoustic model set, and $P(u)$ is the prior probability of u from the language model.

The arcs in the lattices are usually gathered into clusters to form the sausage-like structures as Fig. 2.1b and Fig. 2.1c to make the indexing task easier and reduce memory requirements. The standard text indexers can be directly used for indexing these structures. In addition, because the arcs in the same cluster with the same word hypotheses would be merged, the memory used for the sausage-like structures is less than the lattices [10]. Examples of such sausage-like lattice-based structures include Confusion Networks (CN) [13,14], Position Specific Posterior Lattices (PSPL) [11,15,16], Time Merged Indexing (TMI) [17], and Time-Anchored Lattice Expansion (TALE) [18]. Fig 2.1b is an example of CN structure for the lattice in Fig 2.1a. CN clusters the arcs based on their time spans, and the orders of the arcs in the original lattice are preserved. However, some word sequences included in the original lattice are discarded. For example, the word sequence $\{W_3, W_4, W_5\}$ in Fig 2.1a can not be found in the CN structure in Fig 2.1b. On

the other hand, PSPL in Fig 2.1c not only preserves all the word sequences in the original lattice, but also generates some word sequences not in the original lattice. For instance, the word sequence $\{W_1, W_2, W_5, W_{10}\}$ in the PSPL structure in Fig 2.1c does not exist in Fig 2.1a. The new word sequences generated by PSPL sometimes enhance the retrieval process [17,18]. For more comparison of PSPL and CN, the reader is referred to [19].

Indexing can also be implemented by representing the lattices as weighted automata and building an index for all of the possible sub-strings contained in the lattices [20]. Under this general framework, the index itself is a weighted finite state transducer (WFST) whose inputs are word strings, and the outputs are a list of spoken segments and their relevance scores.

2.3 Out-of-Vocabulary Problem and Subword-based Indexing

The vocabulary is predetermined before the speech is passed to the ASR system for recognition. If a word spoken in the audio is not present in the vocabulary of the recognizer, the recognition system can never correctly recognize that word. Therefore, if a query is out-of-vocabulary (OOV), the retrieval system cannot find the segments containing the query even the retrieval process is conduced on the lattices. Unfortunately, since the less common and topic specific words form the great part of the queries, the percentage of OOV queries can be higher than 15% on a real system [21].

Some people have suggested that the OOV problem can be tackled by building an ASR system generating transcriptions or lattices based on subword units [22–24]. At

the retrieval time, when an OOV query is encountered, the query is converted into a sequence of subword units, and then the system matches the subword unit sequence in the subword-based recognition output. Although this approach conceivably eliminates the OOV problem, the post-recognition indexing and retrieval may become more complex under this approach [25]. For example, grapheme-to-phoneme technique is usually needed to transform a word into a subword sequence [26,27].

A wide range of subword units can be considered, which can be roughly divided into two categories, *linguistically motivated units* as well as *data-driven units*. The linguistically motivated units include syllable [19,22], phoneme [28], or subphone units [29]. Linguistically motivated units require knowledge about specific language and may be costly to extract for some languages. Data driven units are derived by utilizing statistical and information theoretic principles. Phone multigrams [30] are non-overlapping and variable-length phone sub-sequences with predefined maximum length. These are found using an unsupervised iterative algorithm maximizing the likelihood of the training data under the multigram language models. Similarly, particles [31] are selected in a greedy fashion so as to maximize the leave-one-out likelihood of a bigram language model. Statistical morphs [32,33] are based on the minimum description length (MDL) principle, which means that in addition to the corpus representation given by the data likelihood, the lexicon representation is also taken into account. Graphemes (or letters) have also been proposed as subword units for spoken content retrieval [34]. In such system, the grapheme-to-phoneme module is not needed.

Word-based indexing and subword-based indexing have different strengths and weakness. Word-based approaches suffer from OOV words and as a result have lower recall.

Subword-based approaches result in higher recall at the potential expense of lower precision. Hence a combination of different units may yield the best performance. One way to achieve this is using both word and subword indices for retrieval [22,35,36]. For example, when a query is entered, the system first individually generates the results based on word and subword indices, and then combines the results from different indices via simply integrating their relevance scores. This approach requires determining some parameters such as interpolation weights, which can be learned by the learning-to-rank techniques from a set of training queries [37].

2.4 Query-by-Example

In *query-by-example* [38–43], instead of providing a query in text form, the user provides one or more audio examples of the query. These audio examples could be found by the user in a data pool, or even directly spoken by the user.

Although query-by-example has become a topic of recent interest, it has its roots in the early template-based approaches to speech recognition. Before statistical methods become the predominant approaches to speech recognition, early speech recognizers often employed Dynamic Time Warning (DTW) search mechanisms which relied on direct acoustic similarity measures between stored templates and test data. Although acoustic similarity measures often suffer from mismatches due to speaker, channel, or environments, direct similarity measures were recently used with some success for query-by-example [44]. Alternatively, some recent works has examined to use phone posteriorgram or Guassian posteriorgram for template matching [39,40]. Statistical models such as HMM are helpful to query-by-example [45,46].

2.5 Evaluation Metrics

Precision, Recall and F-measure

Precision, Recall and F-measure are relatively familiar metrics. Precision is the fraction of relevant objects in the retrieved objects, and recall is the fraction of relevant objects retrieved. F-measure is

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

Actual Term Weighted Value (ATWV)

Actual Term Weighted Value (ATWV) is defined in the NIST STD 2006 Evaluation Plan [47] which is only used in STD.

$$ATWV = 1 - \frac{1}{|Q|} \sum_{all\ queries\ Q} \{P_{miss}(Q) + \beta P_{FA}(Q)\}, \tag{2.5}$$

where β is a user-defined parameter (set to 1000 in the 2006 NIST STD evaluation), and $|Q|$ is the number of testing queries.

$$P_{miss}(Q) = 1 - \frac{C(Q)}{R(Q)},\ P_{FA}(Q) = \frac{A(Q) - C(Q)}{T - C(Q)}, \tag{2.6}$$

with T being the total duration of the speech in the collection. Here $R(Q)$ is the total number of times the specific query Q actually appears in the audio collection, $A(Q)$ is the number of hits returned when query Q is requested, and $C(Q)$ is the number of hits that are actually correct.

The term "actual" in ATWV refers that the system should automatically decide a threshold which determines whether to return a hit or not. If the system tunes the threshold to maximize (2.5), then we obtain Maximum Term-Weighted Value (MTWV). Moreover,

if we weight each query by its occurrence times in the audio collections (the queries appear frequently in the corpus may have larger probabilities to be requested), then we have Actual Occurrence-Weighted Value.

Precision@N, R-precision, Mean Average Precision (MAP)

The evaluation metrics just introduced do not consider the ranking of the retrieved objects. Here in this subsection some evaluation metrics for the ranking results are introduced.

Precision@N is the precision measure of the top N returned objects. R-precision is similar to precision@N, except that N varies for each given query Q and is set to the total number of relevant objects for Q in the collection.

Mean Average Precision (MAP) [48] is the **mean** of the *Average Precision* over the testing queries. The average precision for the retrieval result of a query is defined as (2.7),

$$Average\ Precision = \frac{\sum_{k=1}^{n} precision(k) rel(k)}{R}, \tag{2.7}$$

where R is the number of relevant objects for the query, n is the total number of objects in the ranking list, $precision(k)$ is the precision measure of the top k objects in the list (that is, Precision@k), and $rel(k)$ is an indicator function which equals one if the item at rank k is a relevant object, and zero otherwise. The value of MAP can also be understood as the area under the precision-recall curve. Fig 2.2 is an example for demonstrating the computation of average precision. Suppose the ranking list in the middle of Fig 2.2 is returned by the retrieval system, in which blue balls represent relevant objects, and the red ones are irrelevant objects. Since there are four relevant objects with precisions 1.00, 1.00, 0.75, 0.50 respectively in the ranking list, average precision of the ranking list is

$$Average\ Precision = \frac{1}{4}(1.00 + 1.00 + 0.75 + 0.50) = 0.8125 \tag{2.8}$$

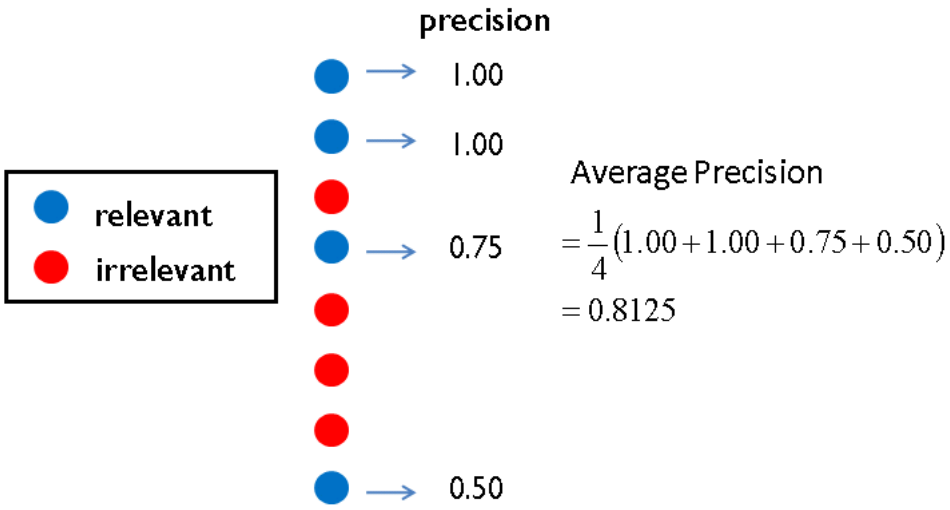


Figure 2.2: Example demonstrating the computation of Average Precision (AP).

In this thesis, MAP is used for evaluating the performance of the results in all of the experiments.

2.6 Optimizing Evaluation Performance

One recent trend in spoken content retrieval has focused on efforts to directly optimize systems to their evaluation metrics. A linear model whose parameters are learned to directly maximize figure of merit using gradient descent algorithm was introduced to transform the phone posterior probability output for STD [49]. STD evaluation metrics have also been used recently to optimize the weights on the indexing features (terms in a query) to reflect their importance for different contexts in which the query term occurs [50]. User feedback [51] is also integrated into a discriminative training process in order to optimize retrieval performance. The posterior probabilities derived from lattices are enhanced to minimize a loss function that boosts the relevance scores of the correctly retrieved segments over the competing incorrect ones via user feedback obtained from initial retrieved

results.

2.7 Machine Learning Methods

There have been some works [8,52–55] taking advantage of the discriminative capability of machine learning methods such as support vector machines (SVM) or multi-layer perceptrons (MLP) to facilitate STD. In those works, a set of training data (a set of queries and associated relevant/irrelevant segments) is assumed available. Then SVM or MLP is used for training a model which identifying if a spoken segment contains the entered query term or not. In order to have these machine learning classifiers properly work for the target spoken archive, the training data must be reasonably matched to the target spoken archive, but such data is usually not available or difficult to collect. In fact, in reality it is very possible that the spoken archive includes content produced in different parts of the world by different speakers on different domains under different acoustic conditions. This makes collecting a reasonably good training set very difficult. Moreover, since the characteristics of the queries are usually very diverse, a simple classifier optimized for many different training queries may not be able to offer the best solution for the high variety of many different testing queries.

2.8 Benchmark Data Sets

TREC SDR

The TREC SDR evaluations focused on a corpus of broadcast news speech from various sources including CNN, ABC, PRI and Voice of America. About 550 hours of speech

were segmented manually into 21574 stories. ASR systems tuned to the broadcast news domain had 15-20% word error rates, and retrieval based on the ASR output achieved very similar performances compared with approximate manual transcripts. As a result, NIST's final report on the TREC SDR evaluations declared the research effort "a success story" [3].

NIST STD

The National Institute of Standards and Technology (NIST) STD 2006 Evaluations [47] introduced the task of locating the exact occurrence of a query in large heterogeneous speech archives including broadcast news, telephone conversations and meetings. The corpus used for the evaluation included Arabic, Mandarin Chinese and English, and attracted many different sites [7–9].

NTCIR SDR

Recently NTCIR has a spoken content retrieval track [56]. There are two subtasks in this track, STD and Spoken Document Retrieval (SDR). Both of the subtasks target to retrieve academic and simulated lectures of the Corpus of Spontaneous Japanese (CSJ), which includes 2702 lectures (about 600 hours). The STD task is to find all spoken segments[2] that include a specified query term in CSJ. For SDR, the retrieval system should find the passages including the relevant information related to the query.

[2]They are called Inter-Pausal Unit (IPU) in the task description.

2.9 Spoken Content Retrieval in the Real World

While this chapter has largely focused on the technologies about spoken content retrieval, it is important not to overlook its applications in the real world. In this section, some well-known on-line systems based on the techniques of spoken content retrieval are introduced.

SpeechBot

SpeechBot [57] was a program started by HP Labs in 1999 to index and make searchable audio and video programs using speech recognition software. The service once had 15,321 hours of content available for search before SpeechBot was shut down after HP closed their Cambridge research lab.

SpeechFind

SpeechFind [58] system is a spoken document retrieval system currently serving as the search engine for the National Gallery of the Spoken Word (NGSW) [59]. The speech corpus from NGSW covers one of the largest ranges of a audio material available today up to 60,000 hours from the last 110 years. The audio content includes a diverse range of vocabulary over the time periods. Many of these include various kinds of acoustic conditions (e.g., background noise, reverberation, channels, recording media, speaking style, etc.)

PodCastle

PodCastle [60,61] is a service that enables the searching of speech data such as podcasts, individual audio or video files on the web, and video clips on video sharing services (Nico

Figure 2.3: The error correction interface of Podcastle.

Nico Douga, YouTube, and Ustream). The most special part for PodCastle is it allows the users accessing the PodCastle service to correct speech recognition errors, and the system uses the correction information for recognition model re-estimation.

GAudi

In the 2008 presidential election race in the United States, the prospective candidates made extensive use of YouTube to post video material. Google developed a scalable system, GAudi (short for Google Audio Indexing), which transcribes this material and makes the content searchable (by indexing the meta-data and transcripts of the videos) [62]. GAudi was once available on the Web at labs.google.com [63].

MIT Lecture Browser

MIT has released a new search engine which lets users search MIT audio and video lectures [64]. The browser enables the user to type a text query and receive a list of hits contained within the indexed lectures. Queries can be constrained by allowing users to specify a topic category from a pull-down menu before searching. Also, the speech recog-

nition output could be manually corrected to improve the transcriptions [65,66].

NTU Virtual Instructor

NTU Virtual Instructor [67,68] manages lecture courses offered in National Taiwan University (NTU). Its functionalities include spoken content retrieval, topic segmentation, key term extraction, hierarchical summarization, semantic structuring for the courses, and key term graph construction. The user can ask questions to the system and learn what he needs in his own way.

Fig. 2.4 is the retrieval interface of NTU Virtual Instructor. After the user enters a text query (which is "Viterbi" in Fig. 2.4), a list of relevant spoken segments is retrieved and displayed with the *Play* bottoms. The user can select to only listen to the individual segments or the whole classes containing the retrieved segments. Since most courses were produced by the instructors primarily in Mandarin Chinese but embedded with some English terminologies, the system supports both Chinese and English queries. Moreover, the retrieval module in NTU Virtual Instructor operates on the lattice-based indices.

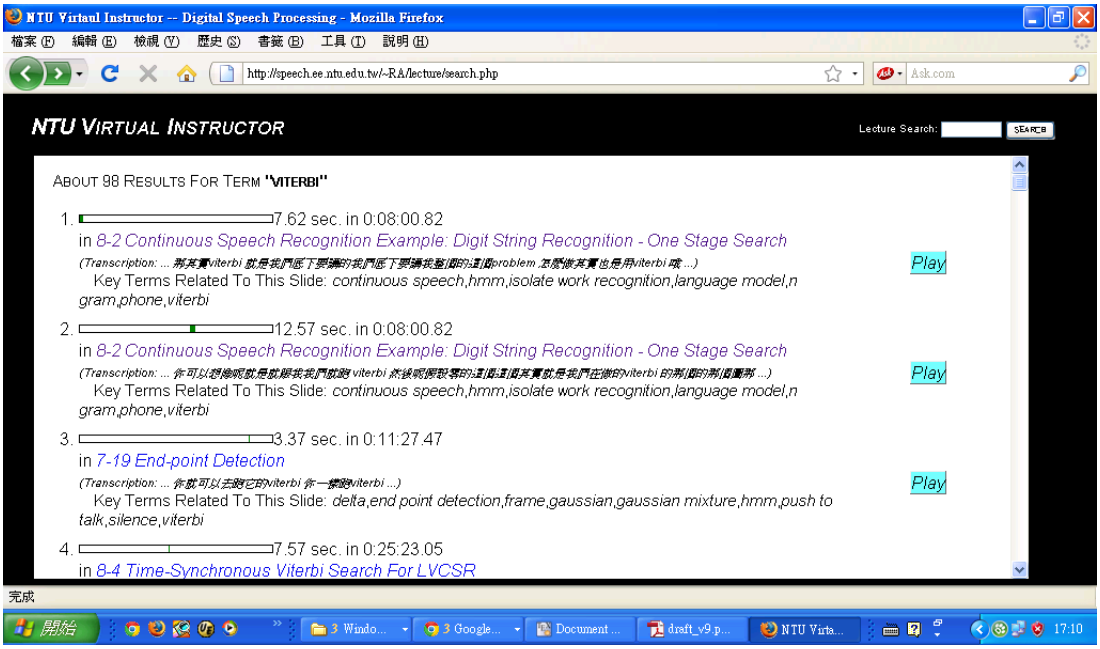


Figure 2.4: A screen shot of NTU Virtual Instructor.

Chapter 3 Relevance Feedback

Since the approaches proposed in this thesis are all based on the scenarios of relevance feedback, in this chapter we introduce the relevance feedback scenarios considered in this thesis. Relevance feedback is a mature technique for text information retrieval [69], and it has been applied on numerous popular text retrieval models such as the vector space model [70], the probability model [71], and the language model [72]. It has been used extensively in different retrieval domains such as image [73–78] and video retrieval [79–81]; however, it has not yet been fully leveraged for speech information retrieval. Relevance feedback can also be used with the learning-to-rank techniques. Learn-to-rank is a set of techniques learning retrieval models for object ranking based on a set of training data usually labelled by the experts. However, since hiring experts for data labelling is very expensive, relevance feedback can be served as the approach for training data collection [82,83].

As mentioned in Section 2.1, when a query Q is entered, the spoken content retrieval system ranks the returned objects x based on the values of a relevance score function $S(Q, x)$ evaluated for x with respect to Q. The basic idea for relevance feedback is to modify the original relevance score function $S(Q, x)$ to yield a better function $S'(Q, x)$ via relevance information obtained from the feedback loop. There are in general two different scenarios of relevance feedback: *user relevance feedback* and *pseudo-relevance feedback*. These are further discussed in the following.

3.1 User Relevance Feedback

As implied by the name, in the scenario of user relevance feedback, the users provide relevance information for improving the system performance at the search phase. It has been known for long that involving the users into the search process by relevance feedback is an effective way of improving the retrieval performance. During the search process, the user provides information about relevant segments as positive examples and irrelevant segments as negative examples with respect to the query entered. Thus the system learns from these examples to yield improved performance.

Although in this scenario the relevance information comes from the user, implicit feedback [84–91] has been widely used in real systems because most users are reluctant to give relevance feedback explicitly. Implicit feedback means the system analyses the user's behaviour on-line to get the feedback information; the user does not know he is in a feedback procedure. One representative is click-through data [82,85]. Web search engine like Google usually returns not only a list of web page links but also the abstracts of the pages. Because a user can decide if the web pages retrieved are relevant via viewing the abstracts, it may be conceivable to assume that the user only clicks on the web pages considered relevant. Thus, if a user clicks the the third link on the returned list without clicking on the first two, it is reasonable for the system to assume that the third web page is relevant and that the first two are irrelevant. For spoken content retrieval, we can assume that the transcription is displayed beside each spoken document or segment on the returned list given by the retrieval system, and then the user is able to judge if the spoken document or segment is what he wants based on the automatic transcription (Human comprehension of error corrupted transcripts is generally not degraded for low

enough error rates, and the identification of the general topic is possible even for higher error rates [92]).

There are two scenarios, *short-* and *long-term context*, for user relevance feedback [93]. For short-term context user relevance feedback, the retrieval system obtains only the relevance information for the single query a user just entered, and the relevance feedback process aims at only improving the retrieval performance for exactly the current entered query. For example, a user is looking for relevant objects about the query "White House". During his search process, if the system is given some relevant and irrelevant objects with respect to the query "White House", then the retrieval system adjusts its relevance score function to improve the retrieval results for the query "White House" based on the relevance information. For long-term context user relevance feedback, the historical record of relevance information for many different queries is collectively used to improve the retrieval performance over all other queries. For instance, some users have looked through the results returned by the system with respect to the queries "US", "Amarican", "Obama" and so on, and they gave the system some feedback during their search processes. Those feedback information was recorded and used to generally improve the system performance for all the queries. Therefore, the retrieval results for the query "White House" may also be improved even if no user gave feedback with respect to "White House" before. Those two scenarios are further discussed below.

3.1.1 Short-term Context User Relevance Feedback

Fig. 3.1 (a) shows short-term context user relevance feedback for spoken content retrieval. The user browses the retrieved list on the left side ranked by the original score $S(Q, x)$.

If the user gives the relevance information of the top N spoken segments on the list to the system [1], for example, in Fig. 3.1 (a) he selects items 1 and 3 as relevant but item 2 as irrelevant, those labelled segments are used to obtain the new score $S'(Q, x)$, which is used to re-rank the spoken segments below the top N. Note that the order of these top N labelled spoken segments should be frozen [38,69]. In practice, the returned results are usually divided into pages. When the user clicks through the first page, he actually gives relevance information implicitly to the system. When he starts to browse the second page, the system has already changed the ranking order of the spoken segments after the first page based on the new score which includes the relevance information from the first page. In this case, the top N spoken segments with user relevance information are the spoken segments in the first page, and because the user has already seen them, re-ranking their order is meaningless, and thus they should be frozen.

3.1.2 Long-term Context User Relevance Feedback

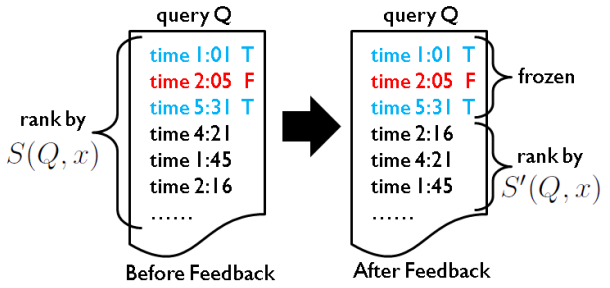
Fig. 3.1 (b) shows long-term context user relevance feedback for spoken content retrieval. Historical relevance information for many queries (training queries such as Q_1, Q_2, and Q_3 in Fig. 3.1 (b)) entered by one or more users is collectively used to train the new score, $S'(Q, x)$, which is used to rank the spoken segments corresponding to the new query Q'.

[1] Because the user usually browses the retrieved objects based on the ranking orders provided by the system, the assumption that the user labels the top N objects on the returned list is quite common in the literatures [38,94–98].

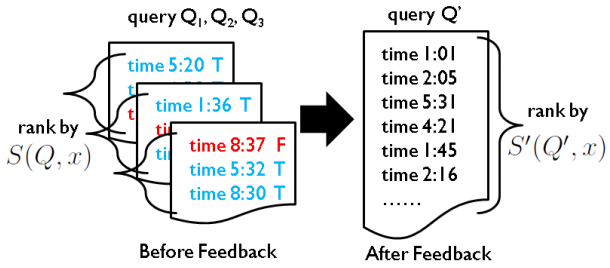
3.2 Pseudo-Relevance Feedback

Pseudo-relevance feedback (PRF), also known as blind relevance feedback, has been widely used for long in information retrieval to obtain relevance information for each query without actually involving any action from the user. It has been successfully applied on different retrieval domains like text [99–107], image [108] and video [79,109]. Conventionally, PRF assumes that a small number of top-ranked objects in the first-pass retrieved results are relevant (or "pseudo-relevant"), and sometimes in addition some bottom-ranked objects are irrelevant (or "pseudo-irrelevant"), and these pseudo-relevant (and -irrelevant) objects can then be taken as extra information to improve the retrieval results.

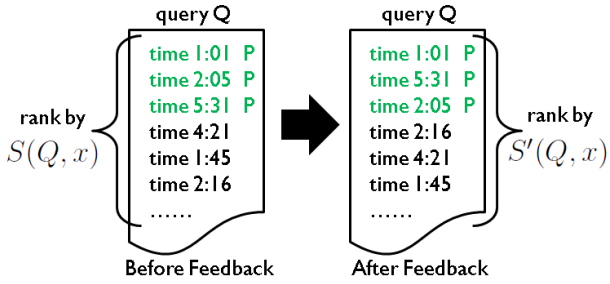
Fig. 3.1 (c) shows pseudo-relevance feedback for spoken content retrieval. The system simply assumes the top three spoken segments in the first-pass returned list ranked by $S(Q, x)$ are relevant without any user input; these pseudo-relevance segments are then used as positive examples to obtain $S'(Q, x)$ as in the short-term context. All of the returned spoken segments are then re-ranked based on this new score. Note that no spoken segments' orders should be frozen here because no spoken segment has been labeled by the user. In fact, what is presented to the user is the re-ranked list of spoken segments after pseudo-relevance feedback.



(a) short-term context user relevance feedback



(b) long-term context user relevance feedback



(c) pseudo-relevance feedback

Figure 3.1: Different relevance feedback scenarios for spoken content retrieval. The original score $S(Q,x)$ before relevance feedback is changed to $S'(Q,x)$ after relevance feedback. Spoken segments with **T**, **F** and **P** are respectively the user-labeled relevant and irrelevant segments, and those assumed relevant by the system.

Part II

Improved Spoken Content Retrieval

Chapter 4 Retrieval Oriented Acoustic Model Re-estimation by Relevance Feedback

The goal for the Spoken Term Detection (STD) system is to return a list of spoken segments which include the query term the user enters. The segments include the query term are considered as "relevant" in this task. In a typical STD system, the target spoken archive is first segmented into spoken segments, and then each spoken segment x is transcribed into a lattice. After the user enters a text query Q, the retrieval engine evaluates the degree of relevance for each segment x with respect to the query Q, represented by relevance score function $S(Q, x)$ which is usually derived from the lattice of x. Then the system returns the segments whose $S(Q, x)$ are larger than a threshold. and ranked the results according to $S(Q, x)$.

As mentioned in Chapter 1, people usually consider recognition and retrieval as two cascaded independent modules for STD, and the retrieval techniques were usually assumed to be applied on top of some ASR output. In this chapter, I propose a new framework: to integrate the two parts into a single task. This can be achieved by adjusting the acoustic model parameters borrowing the techniques of discriminative training based on user relevance feedback. The modified acoustic models then give updated relevance score functions for STD.

4.1 Introduction

The relevance score function $S(Q, x)$ is typically lattice-derived and depends on the acoustic model parameters θ used for generating the lattices thereby. Therefore, the complete relevance score function $S(Q, x)$ should include the acoustic model parameters θ:

$$S(Q, x|\theta) = \frac{\sum_{u \in W(x)} P_\theta(x|u)P(u)N(u, Q)}{\sum_{u \in W(x)} P_\theta(x|u)P(u)}, \tag{4.1}$$

where $W(x)$ is the lattice of segment x, u is an allowed word sequence in the lattice $W(x)$, $P_\theta(x|u)$ is the likelihood for observation sequence x given the word sequence u based on the acoustic model set θ, $P(u)$ is the prior probability of u from the language model, and $N(u, Q)$ is the occurrence count of query Q in u. Since the denominator in (4.1) is the sum of the likelihoods of all word sequences u in the lattice, and the numerator of (4.1) is the same but weighted by the occurrence count of query Q, equation (4.1) is the expected occurrence count of query Q in lattice $W(x)$ based on the set of acoustic models θ.

If the user gives some feedback to the system, for example he/she selects items 1 and 3 shown in Fig. 4.1 as relevant but item 2 as irrelevant, a new set of acoustic models θ^* can then be estimated based on the feedback and so on. Thus, because the relevance score function in (4.1) depends on the acoustic model parameters, it is changed into $S(Q, x|\theta^*)$ accordingly, which in turn yields new ranking results. This technique can be very helpful for a search engine aiming at indexing spoken segments available on many web sites over the Internet with various acoustic/linguistic conditions, for which adapting the acoustic/language models for the various acoustic/linguistic conditions is almost impossible. With this approach proposed here, acoustic models can be adjusted based on user relevance feedback. This can be an important step towards a more robust spoken content retrieval technologies.

Some research works have in fact considered the recognition and retrieval process as a whole to try to improve retrieval performance, but efforts deliberating on the interaction between the recognition and retrieval processes are still very limited. One good example is considering the recognition error pattern with a confusion matrix during retrieval [51,110,111]. This involves inferring the correct words actually appearing in the spoken segments from the erroneous ASR transcriptions. Some have also observed that although word accuracy is an excellent metric for recognition performance, it is not directly related to retrieval performance [112–114]. For example, words frequently used as query terms should be correctly recognized, while recognition errors for function words have almost no impact on retrieval performance. As a result, word significance has been taken into account during decoding [112,113], and a minimum classification error (MCE [115]) discriminative training method was used that also took into account word significance [114]. In another approach, when an OOV query term is entered, the OOV term is dynamically inserted into the possible positions in the lattice to handle the OOV query [116].

On the other hand, estimating acoustic model parameters based on pre-defined criterion is a well-studied problem in speech recognition. Applied to STD with relevance feedback, however, the problem is different from the conventional acoustic model training approaches for speech recognition in at least two ways:

1. The system input includes only whether a spoken segment is relevant to a query or not; it does not include the transcription of any utterance [60,61].

2. The goal is to improve retrieval performance rather than recognition accuracy.

In the following sections, a set of objective functions that take into account the retrieval

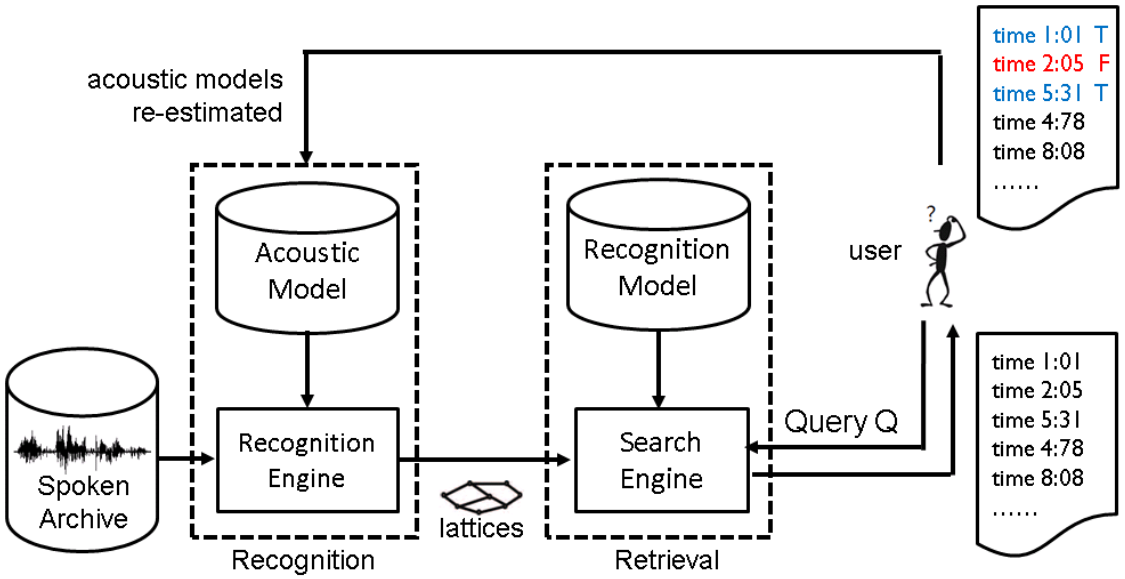


Figure 4.1: The framework of the proposed approach.

process as well as discriminative training algorithms that optimize these objective functions is developed [49,117–119].

4.2 Scenario

The acoustic model re-estimation methods can be used in short-term (Section 3.1.1) and long-term (Section 3.1.2) contexts user relevance feedback in Section 3.1, and pseudo-relevance feedback (PRF) in Section 3.2 as well.

In the scenario of short-term context user relevance feedback, after a query is entered, the retrieval system performs the first-pass retrieval and returns a list of spoken segments. Some segments in the returned list are then labelled as relevant or irrelevant by the user and taken as training data. The parameters of a new set of acoustic models are then estimated on-line according to the feedback information. The first-pass returned list is finally re-ranked based on the new set of acoustic models. In this scenario, the acoustic

models are trained based on the feedback information for the query entered, with a goal to improve the retrieval performance for the specific query. Because the training data is very limited, it is possible to perform the acoustic model training and re-ranking of returned list on-line. The scenario of PRF is exactly the same as short-term context user relevance feedback, except that the relevance information is automatically assumed by the system.

In the scenario of long-term context user relevance feedback, the system collects the feedback information of a set of queries entered in a certain period of time to estimate a new set of acoustic models. Then the lattices of the whole corpus to be retrieved are re-scored based on the new set of acoustic models with a goal to improve the retrieval performance for queries to be entered in the future.

4.3 Acoustic Model Re-estimation in Short-term Context User Relevance Feedback

4.3.1 Objective Function

Given positive and negative (or relevant and irrelevant) examples for a certain query Q from the user relevance feedback, the system estimates a new set of acoustic model parameters θ^* by maximizing an objective function $F(\theta)$,

$$\theta^* = \arg\max_{\theta} F(\theta). \tag{4.2}$$

With the new set of acoustic models, the likelihood $P_\theta(x|u)$ in (4.1) is replaced by $P_{\theta^*}(x|u)$, so the original relevance score function in (4.1) for each segment is modified accordingly to $S(Q, x|\theta^*)$, based on which all the segments in the first-pass returned list

are re-ranked. The above procedure is conducted on-line. It is not very time consuming because only a limited amount of data is used for model training. The new acoustic models are stored only in memory and are discarded after the retrieval session. Several objective functions $F(\theta)$ in (4.2) are proposed below. The objective functions described below can certainly be applied in the scenario of PRF, except that the positive and negative examples for training are derived by the system.

Basic Forms

The first objective function $F_1^Q(\theta)$ to be maximized in (4.2) is the sum of the relevance scores of all positive examples

$$F_1^Q(\theta) = \sum_{x_t^Q} S(Q, x_t^Q|\theta), \tag{4.3}$$

where x_t^Q is a positive example with respect to the query Q. The second objective function $F_2^Q(\theta)$ is then the sum of the distances between all positive and negative example pairs,

$$F_2^Q(\theta) = \sum_{x_t^Q, x_f^Q} [S(Q, x_t^Q|\theta) - S(Q, x_f^Q|\theta)], \tag{4.4}$$

where x_f^Q is a negative example with respect to the query Q. The new acoustic models θ^* obtained by maximizing (4.4) best separating the relevance scores between positive and negative examples.

Considering Evaluation Measures

Since Mean Average Precision (MAP) widely used in many STD tasks is used here as the basic measure to evaluate retrieval performance in the experiments, maximizing the distances between all pairs of positive/negative examples as in (4.4) does not necessarily yield improved retrieval performance. MAP quantifies the goodness of the ranked

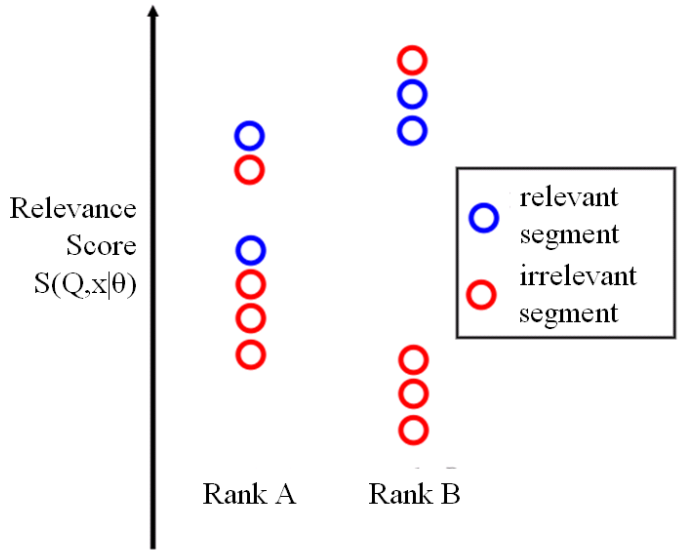


Figure 4.2: Rank B leads to larger $F_2^Q(\theta)$ in (4.4), but rank A is a better ranking in terms of MAP.

retrieval results, and as such favours retrieval results with relevant objects ranked higher than irrelevant objects. Thus the relative levels of all positive examples with respect to all negative examples are more important than their individual absolute relevance score differences. To be specific, if a positive example already has a higher relevance score than all negative examples, any increase in the relevance score of this positive example cannot further benefit retrieval performance. Therefore, the second objective function $F_2^Q(\theta)$ is not effective enough. Fig 4.2 is the example further demonstrating the above statement. Each circle represents a spoken segment. The blue circles are relevant segments, and the red ones are irrelevant segments. The vertical scale in Fig 4.2 is the relevance scores of the spoken segments. In terms of $F_2^Q(\theta)$ in (4.4), rank B is better than rank A since the relevant and irrelevant segments are better separated in rank B. However, rank A is a better ranking because in rank A the irrelevant segments only exceed one relevant segment, but in rank B both relevant segments are surpassed. Thus actually the MAP value for rank

A is larger than rank B.

The acoustic model training process can be enhanced if we can estimate a set of acoustic models that directly maximizes the MAP of the training examples. Although directly optimizing MAP may be difficult, it has been found that maximizing an accuracy count $A(\theta)$ is equivalent to maximizing a lower MAP bound [82,120]:

$$A(\theta) = \sum_{x_t^Q, x_f^Q} \delta(x_t^Q, x_f^Q), \tag{4.5}$$

where

$$\delta(x_t^Q, x_f^Q) = \begin{cases} 1 & S(Q, x_f^Q|\theta) > S(Q, x_f^Q|\theta) \\ 0 & otherwise \end{cases}. \tag{4.6}$$

$A(\theta)$ hence represents the number of positive/negative example pairs in which the relevance of the positive example is greater than that of the negative example. However, since $A(\theta)$ in (4.5) is not differentiable, it is not easily optimized. Therefore we approximate $\delta(x_t^Q, x_f^Q)$ in (4.6) with

$$sigmoid(x_t^Q, x_f^Q) = \frac{1}{1 + e^{c[S(Q,x_f^Q|\theta) - S(Q,x_f^Q|\theta)]}}, \tag{4.7}$$

and define the third objective function to be optimized as

$$F_3^Q(\theta) = \sum_{x_t^Q, x_f^Q} sigmoid(x_t^Q, x_f^Q). \tag{4.8}$$

In (4.7), as $S(Q, x_f^Q|\theta)$ is larger than $S(Q, x_f^Q|\theta)$, $sigmoid(x_t^Q, x_f^Q)$ tends to 1; otherwise, $sigmoid(x_t^Q, x_f^Q)$ tends to 0. c is a constant that controls the slope of the sigmoid function.

Considering Unlabelled Data

When utilizing (4.8) as the objective function, the estimated acoustic model may overfit to the training examples. For instance, the acoustic models may rank all positive examples

higher than all negative examples, but it is possible that some positive examples may be scored lower than some unlabelled segments, since the unlabelled data is not considered at all in (4.8). This may not be good because some of these unlabelled segments may be irrelevant. Hence, we wish to estimate a set of acoustic models which keeps the positive examples ranked at the top of the first-pass returned list, including those unlabelled, to prevent such overfitting. This can be achieved by replacing the objective function $F_3^Q(\theta)$ with

$$\begin{aligned} F_4^Q(\theta) &= F_3^Q(\theta) \\ &+ \rho \sum_{x_t^Q, x_{un}^Q} sigmoid(x_t^Q, x_{un}^Q), \end{aligned} \tag{4.9}$$

where x_{un}^Q is an unlabelled segment within the returned list and ρ is a weighting parameter. $sigmoid(x_t^Q, x_{un}^Q)$ tends to 1 if x_t^Q has a higher relevance score than x_{un}^Q. Equation (4.9) can be viewed as a smoothing approach that ensures the unlabelled segments are given lower scores than the positive examples. From another point of view, since each query only has few relevant segments, in general most retrieved segments are irrelevant, so it is reasonable to assume the unlabelled segments as negative examples.

4.3.2 Optimization

All the objective functions presented in Section 4.3.1 can be optimized using the weak-sense auxiliary function similar to that in minimum phone error (MPE) discriminative training [121]. MPE maximizes the expected phone accuracy as

$$F_{MPE}(\theta) = \sum_{r=1}^{R} \frac{\sum_{u \in W(x_r)} P_\theta(x_r|u) P(u) A(u)}{\sum_{u \in W(x_r)} P_\theta(x_r|u) P(u)}, \tag{4.10}$$

where x_r is the r-th training utterance, R is the total number of training utterances, $A(u)$ is the phone accuracy evaluated for the corresponding phone sequence of the word sequence u, while everything else has the same definition as in (4.1). Taking $F_2^Q(\theta)$ in (4.4) as an example, here we first show that objective functions $F_1^Q(\theta)$ and $F_2^Q(\theta)$ mentioned in Section 4.3.1 can be manipulated to have the same form as (4.10) except for a word sequence u with a different definition of $A(u)$.

Recall that the relevance score function in (4.1) is written as

$$S(Q, x|\theta) = \frac{\sum_{u \in W(x)} P_\theta(x|u)P(u)N(u,Q)}{\sum_{u \in W(x)} P_\theta(x|u)P(u)}, \tag{4.11}$$

where $N(u, Q)$ is the occurrence count of the word hypothesis Q in the word sequence u. Hence substituting (4.11) into (4.4) yields

$$\begin{aligned} F_2^Q(\theta) &= \sum_{x_t^Q} \frac{\sum_{u \in W(x_t^Q)} P_\theta(x_t^Q|u)P(u)|x_f^Q|N(u,Q)}{\sum_{u \in W(x_t^Q)} P_\theta(x_t^Q|u)P(u)} \\ &+ \sum_{x_f^Q} \frac{\sum_{u \in W(x_f^Q)} P_\theta(x_f^Q|u)P(u)|x_t^Q|N'(u,Q)}{\sum_{u \in W(x_f^Q)} P_\theta(x_f^Q|u)P(u)} \end{aligned} \tag{4.12}$$

where $W(x_t^Q)$ and $W(x_f^Q)$ are the sets of all possible word sequences in the lattices for the examples x_t^Q and x_f^Q respectively, $|x_t^Q|$ and $|x_f^Q|$ are the total number of positive and negative examples included in the evaluation in (4.4), and $N'(u, Q)$ is defined as $-N(u, Q)$. Therefore, we can optimize (4.12) in exactly the same way as for MPE by simply replacing $A(u)$ in (4.10) by $|x_f^Q|N(u, Q)$ or $|x_t^Q|N'(u, Q)$ as in (4.12). Note that just like in MPE, in the model estimation process, the acoustic models are updated iteratively starting from an initial acoustic model set.

The optimization of $F_3^Q(\theta)$ in (4.8) is more complicated. In the MPE model estimation process, at the i-th iteration, given the acoustic model set θ_{i-1} obtained in the

last iteration, a new acoustic model set θ_i maximizing a weak-sense auxiliary function of (4.10) is estimated. The auxiliary function used in MPE training is

$$\begin{aligned} & H_{MPE}(\theta_i, \theta_{i-1}) \\ & = \sum_{r=1}^{R} \sum_{a \in A(x_r)} \left[\frac{\partial F_{MPE}(\theta_{i-1})}{\partial log P_{\theta_{i-1}}(x_r|a)} \right] log P_{\theta_i}(x_r|a), \end{aligned} \tag{4.13}$$

where $A(x_r)$ represents all the arcs in the lattice of utterance x_r, and $\frac{\partial F_{MPE}(\theta_{i-1})}{\partial log P_{\theta_{i-1}}(x_r|a)}$ is a constant with respect to the acoustic models θ_i to be estimated. $F_3^Q(\theta)$ in (4.8) can be optimized in a similar way. At the i-th training iteration, we find for $F_3^Q(\theta)$ in (4.8) the auxiliary function

$$\begin{aligned} & H_{F_3}(\theta_i, \theta_{i-1}) \\ & = \sum_{x_t^Q} \sum_{a \in A(x_t^Q)} \left[\frac{\partial F_3^Q(\theta_{i-1})}{\partial log P_{\theta_{i-1}}(x_t^Q|a)} \right] log P_{\theta_i}(x_t^Q|a) \\ & + \sum_{x_f^Q} \sum_{a \in A(x_f^Q)} \left[\frac{\partial F_3^Q(\theta_{i-1})}{\partial log P_{\theta_{i-1}}(x_f^Q|a)} \right] log P_{\theta_i}(x_f^Q|a), \end{aligned} \tag{4.14}$$

where $A(x_t^Q)$ and $A(x_f^Q)$ represent all the arcs in the lattices of utterances x_t^Q and x_f^Q. Then the new acoustic model θ_i maximizing (4.14) can be estimated in exactly the same way that (4.13) is maximized in MPE discriminative training. The optimization of $F_4^Q(\theta)$ in (4.9) is then trivial.

4.4 Acoustic Model Re-estimation in Long-term Context User Relevance Feedback

In long-term context user relevance feedback, the system collects a set of training queries $\mathcal{Q}_{train} = \{Q_1, Q_2, Q_3, \ldots\}$ and their positive and negative examples. The retrieval system

can therefore estimate a new set of acoustic model parameters θ^*_{lt} by maximizing

$$F_5^{lt}(\theta) = \sum_{Q \in Q_{train}} F^Q(\theta) \tag{4.15}$$

which is the summation over the objective functions of all the queries in the training query set Q_{train}. F^Q in (4.15) can be $F_1^Q(\theta)$ in (4.3), $F_2^Q(\theta)$ in (4.4), $F_3^Q(\theta)$ in (4.8) or $F_4^Q(\theta)$ in (4.9). The new models θ^*_{lt} are then used to rescore all the lattices in the spoken archive, and then the lattices with new scores are stored and indexed for further use. This approach can yield overall improvements to system performance, even for queries that were not included in the training query set.

4.5 Experiments for Lecture Courses

4.5.1 Experimental Setup

Mean average precision (MAP) was used as the retrieval performance evaluation measure. The pair-wise t-test with a significance level of 0.05 was used to gauge the significance of performance improvements.

33 hours of recorded lectures for a course offered in National Taiwan University was used as the testing spoken archive, and it is quite noisy and spontaneous. The spoken archive was produced by a single instructor primarily in Mandarin Chinese but embedded with some English words. A Chinese lexicon with 10.7K words and a phone set of 35 Mandarin phonemes (NTU-98 [122]) were used. Because of the lack of corpora matched to the topic (technical content of the course) and the style (spontaneous monologue) for the retrieved spoken archive here, the Chinese trigram language model was trained from the Mandarin Giga-word corpus released by Linguistic Data Consortium. Each spoken

segment in the corpus was transcribed into a lattice with beamwidth of 50. 80 Chinese queries were manually selected as testing queries, each consisting of a single word, and another 20 Chinese queries were used as a development set.

In order to evaluate the performance of the proposed approach under different recognition accuracies, we used three sets of acoustic models for generating the lattices:

- Speaker Independent Model (SI): trained by Maximum Likelihood criterion with 4602 state-tied triphones spanned from 35 Mandarin monophones, using a corpus of clean read speech in Mandarin including 24.6 hours of data produced by 100 males and 100 females.

- Speaker Adaptation Model 1 (ADP1): adapted from the SI model with 500 utterances (about 20 minutes) taken from the training set of the lecture corpus mentioned above. Only global MLLR was applied.

- Speaker Adaptation Model 2 (ADP2): adapted from the SI model with 500 utterances taken from the training set of the lecture corpus mentioned above. MLLR with 256 classes cascaded with maximum a posterior estimation was applied.

Since the acoustic models were based on Mandarin phonemes only, the English words embedded in the Chinese utterances were transcribed into Chinese word sequences with similar pronunciation, which made the retrieval task more challenging. The character accuracies of the 1-best transcriptions for the three different sets of acoustic models are shown in Table 4.1.

Table 4.1: Character accuracies for different sets of acoustic models.

	SI	ADP1	ADP2
Character Accuracy	50.26%	62.55%	72.93%

4.5.2 Experimental Results

Here we tested the acoustic model re-estimation approaches in the scenarios of short- and long-term context user relevance feedback and PRF. The acoustic model re-estimation can be started with the SI, ADP1, and ADP2 models used in generating the initial lattices. We assume the correct relevance information for the top N (N = 5,10,15,20) segments were available. The user relevance feedback was used to re-estimate the acoustic model parameters including means, covariances, transition probabilities, and mixture weights.

Short-term Context User Relevance Feedback

Correct relevance information of the top N (N = 5,10,15,20) segments was used here to obtain a new set of acoustic model parameters θ^* as in (4.2). The segments below the top N were then re-ranked based on the new score $S(Q, X|\theta^*)$, while the ranking of the top N segments were frozen. We compared the MAP scores of the returned list before and after re-ranking. All the smoothing parameters in the model training algorithm and the parameter ρ for $F_4^Q(\theta)$ in (4.9) were decided by the development set, and c in (4.7) was set to 1.0.

The experimental results for different objective functions ($F_1^Q(\theta)$, $F_2^Q(\theta)$, $F_3^Q(\theta)$, $F_4^Q(\theta)$ in (4.3), (4.4), (4.8), (4.9)) described in Section 4.3 are shown in Table 4.2, 4.3 and 4.4 with different N (N = 5,10,15,20). SI, ADP1 and ADP2 models were respectively

Number of Feedback Segments	Baseline (Without Feedback)	Objective Functions			
		$F_1^Q(\theta)$	$F_2^Q(\theta)$	$F_3^Q(\theta)$	$F_4^Q(\theta)$
N=5	0.4819	0.4826	$0.5008^{(0)(1)}$	$0.5086^{(0)(1)(2)}$	$0.5106^{(0)(1)(2)}$
N=10		0.4789	$0.5058^{(0)(1)}$	$0.5128^{(0)(1)(2)}$	$0.5140^{(0)(1)(2)}$
N=15		0.4810	$0.5005^{(0)(1)}$	$0.5038^{(0)(1)}$	$0.5044^{(0)(1)(2)}$
N=20		0.4813	$0.4998^{(0)(1)}$	$0.4990^{(0)(1)}$	$0.4998^{(0)(1)}$

Table 4.2: Experimental MAP results for short-term context user relevance feedback with objective functions $F_1^Q(\theta)$, $F_2^Q(\theta)$, $F_3^Q(\theta)$ and $F_4^Q(\theta)$ for N=5,10,15,20. Acoustic model re-estimation is started with the SI models. The superscript labels $^{(0)}$, $^{(1)}$, $^{(2)}$ and $^{(3)}$ respectively indicate significantly better than the baseline, $F_1^Q(\theta)$, $F_2^Q(\theta)$, and $F_3^Q(\theta)$.

considered as the initial models θ in Table 4.2, 4.3 and 4.4. The new model parameter set θ^* was obtained with 3 training iterations. The superscripts labels on the MAP values, $^{(0)}$, $^{(1)}$, $^{(2)}$, and $^{(3)}$ respectively indicate the MAP value is significantly better than the baseline, $F_1^Q(\theta)$, $F_2^Q(\theta)$, and $F_3^Q(\theta)$. Although more user labelled data (more training data) may lead to better acoustic models for the purpose here, the space left for improvements in MAP is reduced. Therefore, increasing the number of feedback segments N did not guarantee more improvements in MAP.

Much can be learned from Table 4.2, 4.3 and 4.4. First, it can be found that $F_2^Q(\theta)$ in (4.4) with the consideration of negative examples was always better than $F_1^Q(\theta)$ in (4.3) except when $N = 20$. Moreover, in all cases $F_2^Q(\theta)$ outperformed the baseline. $F_3^Q(\theta)$ always outperformed the baseline, $F_1^Q(\theta)$, and $F_2^Q(\theta)$ in all cases, except for $N = 20$

Number of Feedback Segments	Baseline (Without Feedback)	Objective Functions			
		$F_1^Q(\theta)$	$F_2^Q(\theta)$	$F_3^Q(\theta)$	$F_4^Q(\theta)$
N=5	0.6189	0.6198	$0.6326^{(0)(1)}$	$0.6326^{(0)(1)}$	$0.6416^{(0)(1)(2)(3)}$
N=10		0.6260	$0.6387^{(0)(1)}$	$0.6426^{(0)(1)(2)}$	$0.6485^{(0)(1)(2)(3)}$
N=15		$0.6286^{(0)}$	$0.6287^{(0)}$	$0.6438^{(0)(1)(2)}$	$0.6427^{(0)(1)(2)}$
N=20		$0.6293^{(0)}$	0.6244	$0.6387^{(0)(1)(2)}$	$0.6399^{(0)(1)(2)}$

Table 4.3: Experimental MAP results for short-term context user relevance feedback with objective functions $F_1^Q(\theta)$, $F_2^Q(\theta)$, $F_3^Q(\theta)$ and $F_4^Q(\theta)$ for N=5,10,15,20. Acoustic model re-estimation is started with the ADP1 models. The superscript labels $^{(0)}$, $^{(1)}$, $^{(2)}$ and $^{(3)}$ respectively indicate significantly better than the baseline, $F_1^Q(\theta)$, $F_2^Q(\theta)$, and $F_3^Q(\theta)$.

for SI models. $F_4^Q(\theta)$ taking into account unlabelled data always outperformed $F_3^Q(\theta)$ in every case, except for $N = 15$ for ADP1 models and $N = 20$ for ADP2 models. $F_4^Q(\theta)$ did not outperform $F_3^Q(\theta)$ in those cases because $F_4^Q(\theta)$ was designed to handle the problem of overfitting, and therefore was of little benefit when N was large. These results in Table 4.2, 4.3 and 4.4 verified that the considerations mentioned in Section 4.3 regarding $F_3^Q(\theta)$ and $F_4^Q(\theta)$ are all correct and contribute to the improvements. $F_4^Q(\theta)$ was found to be the best objective function, and with $F_4^Q(\theta)$ only 5 examples (N=5) were needed to yield very significant improvements over the baseline (0.5106 vs 0.4819 for SI models, 0.6416 vs 0.6189 for ADP1 models, and 0.7504 vs 0.7307 for ADP2 models).

Fig. 4.3 shows the results with different objective functions and different numbers of training iterations when the initial acoustic models were the ADP2 models and $N = 5$.

Number of Feedback Segments	Baseline (Without Feedback)	Objective Functions			
		$F_1^Q(\theta)$	$F_2^Q(\theta)$	$F_3^Q(\theta)$	$F_4^Q(\theta)$
N=5	0.7307	0.7327	0.7366	$0.7443^{(0)(1)(2)}$	$0.7504^{(0)(1)(2)(3)}$
N=10		0.7353	$0.7419^{(0)}$	$0.7431^{(0)(1)}$	$0.7492^{(0)(1)(2)(3)}$
N=15		0.7351	0.7360	$0.7424^{(0)(1)(2)}$	$0.7461^{(0)(1)(2)}$
N=20		0.7382	0.7372	$0.7421^{(0)(1)}$	$0.7416^{(0)(1)(2)}$

Table 4.4: Experimental MAP results for short-term context user relevance feedback with objective functions $F_1^Q(\theta)$, $F_2^Q(\theta)$, $F_3^Q(\theta)$ and $F_4^Q(\theta)$ for N=5,10,15,20. Acoustic model re-estimation is started with the ADP2 models. The superscript labels $^{(0)}$, $^{(1)}$, $^{(2)}$ and $^{(3)}$ respectively indicate significantly better than the baseline, $F_1^Q(\theta)$, $F_2^Q(\theta)$, and $F_3^Q(\theta)$.

The results with 3 iterations in Fig. 4.3 are exactly those listed in a row of Table 4.4. Based on Fig. 4.3 we observed that the results of model re-estimation converged in only a few iterations. For other cases in Table 4.2, 4.3 and 4.4 similar phenomena were also observed; Fig. 4.3 is a typical example. Such results indicate the concept proposed here is practically feasible since the training can be completed quickly on-line.

Long-term Context User Relevance Feedback

In long-term context user relevance feedback experiments, the 80 queries were separated into 2, 4, 8, or 16 folds for cross validation. Each fold was selected once as the testing query set with the other folds set aside as the training query set. For all training queries, we assume the relevance information has been given for top 5 segments (N=5) in the first-

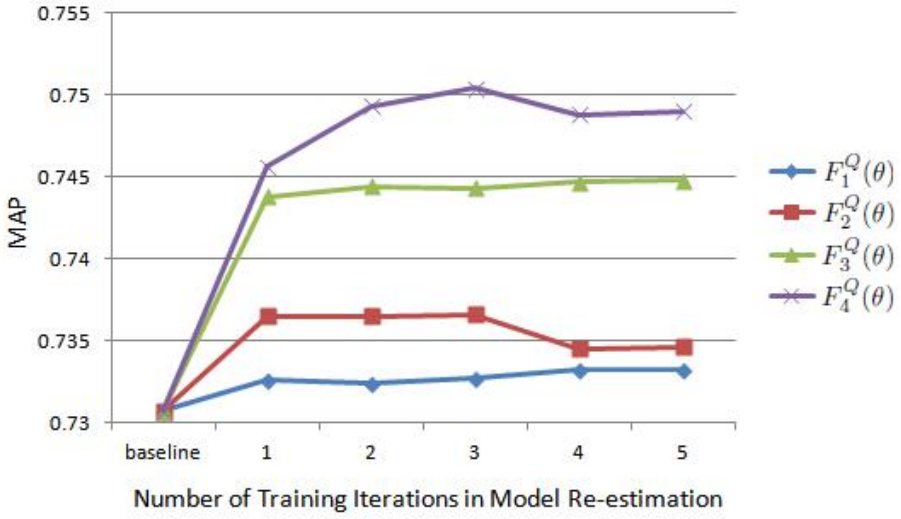


Figure 4.3: Experimental results with different objective functions and different number of training iterations in acoustic model re-estimation when the initial acoustic models were the ADP2 models and $N = 5$ (the relevance information of the top 5 segments were given).

pass returned lists, and we applied $F_5^{lt}(\theta)$ in (4.15) to train a new set of acoustic models using the objective function $F_4^Q(\theta)$. The new acoustic models were used to rescore all the lattices in the spoken archive.

Table 4.5 lists the experimental results with different numbers of training queries. In each test for 2-, 4-, 8-, or 16-fold cross validation respectively 40, 20, 10, or 5 queries were tested, and 40, 60, 70, or 75 queries were used in training. Clearly, the number of training queries affects the performance of the re-estimated acoustic models. In addition, if the acoustic units[1] of a new query do not exist in the training query set, the retrieval performance of the new query may not be influenced by the long-term context relevance feedback; hence the percentage of the acoustic units shared by the training and testing query sets may play an even greater role in the performance of long-term context relevance

[1] triphone models

Cross Validation	Number of Training queries	Acoustic Unit Coverage	Initial Acoustic Models		
			SI	ADP1	ADP2
baseline	-	-	0.4819	0.6189	0.7307
2-fold	40	37%	$0.4999^{(0)}$	$0.6304^{(0)}$	0.7401
4-fold	60	46%	$0.5021^{(0)}$	$0.6386^{(0)}$	0.7410
8-fold	70	47%	$0.5087^{(0)}$	$0.6400^{(0)}$	$0.7444^{(0)}$
16-fold	75	50%	$0.5099^{(0)}$	$0.6419^{(0)}$	$0.7459^{(0)}$

Table 4.5: Experimental results for long-term context user relevance feedback with different numbers of training queries for $N = 5$ (relevance information for top 5 segments were given). Acoustic model re-estimation can be started with the SI, ADP1 or ADP2 models, and the baseline MAPs without relevance feedback are 0.4819, 0.6189, and 0.7307 for lattices generated by the SI, ADP1 and ADP2 models respectively. The superscript labels $^{(0)}$ indicate significantly better than the baseline.

feedback.

The acoustic unit coverage listed in Table 4.5 is the averaged percentage of triphones appearing in the test queries that also appear in the training query set. In Table 4.5, we started acoustic model re-estimation with SI, ADP1 or ADP2 models, and the new acoustic models were used to rescore the lattices generated by the initial acoustic models. Although MAP improvements in general increased with the number of training queries, results showed that it is possible to obtain significant improvements with only 40 training queries each with 5 labeled segments with SI or ADP1 models as the initial models.

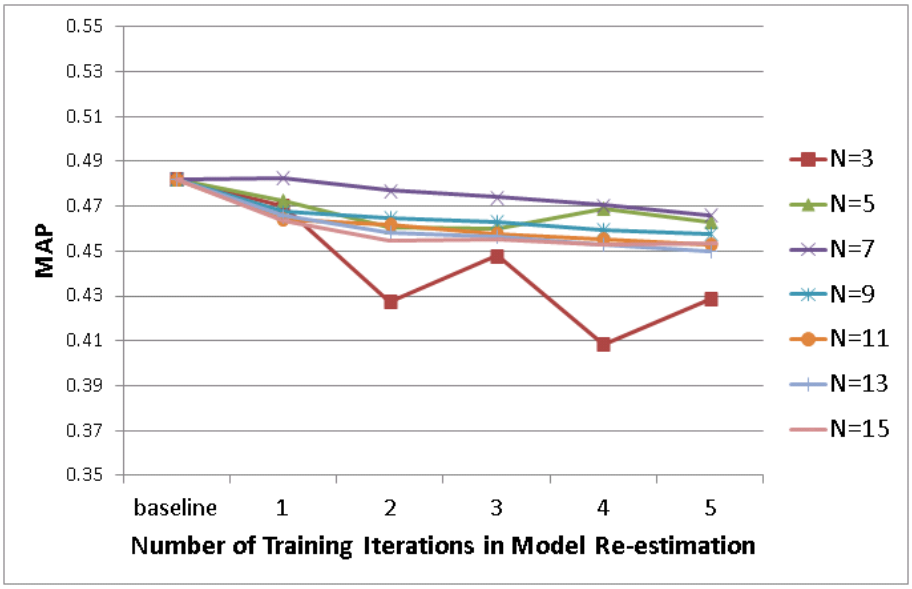


Figure 4.4: Experimental results for PRF which assumed top N segments on the first-pass returned list were positive examples with different number of training iterations in acoustic model re-estimation. The initial acoustic models were the SI models. The objective function $F_1^Q(\theta)$ in (4.3) was used since there were only positive examples.

Pseudo-Relevance Feedback

Fig 4.4 shows the experimental results for PRF which assumed top N segments on the first-pass returned list were positive examples with different number of training iterations in acoustic model re-estimation. The initial acoustic models were the SI models. The objective function $F_1^Q(\theta)$ in (4.3) was used since there were only positive examples. We found that the acoustic model re-estimation method did not offer any improvements in the PRF scenario. There are two possible reasons. First, because in the PRF scenario the segments selected as positive examples were already the ones with the largest relevance scores, the new acoustic models maximizing the relevance scores of these positive examples would not have too much difference from the original models used for generating

the lattices. Second, because there are in general thousands of parameters in the acoustic models, the models learned may be well fitted to the training data. This was fine in the case of user relevance feedback with correct relevance information. However, since there were some incorrect data unavoidably included in the training data (irrelevant segments considered as positive examples), the models fitting the training data "too" well would be misleading by the noisy training examples. Therefore, in the scenario of PRF, due to the noisy training data, training a simpler model with proper regularization may be more suitable than re-estimating the original acoustic models. This will be further discussed in the next chapter.

4.6 Experiments for Broadcast News

In the above experiments, the proposed methods were tested on a set of lecture courses produced by a single speaker. Here the techniques proposed were tested on a broadcast news corpus with many different speakers.

4.6.1 Experimental Setup

A broadcast news corpus in Mandarin Chinese was used as another spoken archive to test the proposed approaches. The news stories were recorded from TV stations in Taipei from 2001 to 2003, with a total length of 198 hours. 160 Chinese queries were manually selected as testing queries, each consisting of a single word. Again, Mean average precision (MAP) was used as the retrieval performance measure, and pair-wise t-test with significance level at 0.05 was also used to test the significance for the performance improvement. The parameters for all the methods were all set to the same values as in the

previous sections without especially mentioned.

For the recognition, 147 right context-dependent initial models plus context-independent final models were used as the acoustic models for simplicity. A tri-gram language model trained on 39M words of Yahoo news, and a set of acoustic models with 64 Gaussian mixtures per state and 3 states per model trained on a corpus of 24.5 hours of broadcast news different from the archive tested here were used. The lexicon contained 60K words. The acoustic vectors used were MFCC with cepstral mean and variance normalization (CMVN) applied. The beam width for recognition was 100. Since 48% and 31% of the speech in the corpus was produced by the reporters and respondents respectively including relatively high background noise, and only 147 acoustic models were used, the character accuracy for the archive was only 54.43%.

4.6.2 Experimental Results

		Number of Feedback Segments (N)			
	Baseline	**N=5**	**N=10**	**N=15**	**N=20**
MAP	0.6302	0.6464*	0.6480*	0.6482*	0.6405*

Table 4.6: Experimental results of broadcast news for short-term context user relevance feedback with objective functions $F_4^Q(\theta)$ for N=5,10,15,20. The superscript label * indicates significantly better than the baseline.

First, the acoustic re-estimation in short-term context relevance feedback in Section 4.3 was tested. Table 4.6 shows the experimental results with different N (N = 5,10,15,20) using the objective function $F_4^Q(\theta)$ in (4.9) which obtained the best perfor-

mance in the experiments of Chapter 4. $F_4^Q(\theta)$ with 3 training iterations were used. Significant improvements over the baseline were observed.

Cross Validation	Number of Training queries	Acoustic Unit Coverage	MAP
Baseline	-	-	0.6302
2-fold	80	87%	0.6319
4-fold	120	93%	0.6340*
8-fold	140	95%	0.6361*
16-fold	150	96%	0.6362*

Table 4.7: Experimental results of broadcast news for long-term context user relevance feedback with different numbers of training queries for $N = 5$ (relevance information for top 5 segments was given). The superscript labels * indicate significantly better than the baseline.

The long-term context user relevance feedback introduced in Section 4.4 was then tested. The 160 queries were separated into 2, 4, 8, or 16 folds for cross validation. Each fold was selected once as the testing query set with the other folds set aside as the training query set. For all training queries, we assume the relevance information has been given for top 5 segments (N=5) in the first-pass returned lists, and we applied $F_5^{lt}(\theta)$ in (4.15) to train a new set of acoustic models using the objective function $F_4^Q(\theta)$.

Table 4.7 lists the experimental results with different numbers of training queries. In each test for 2-, 4-, 8-, or 16-fold cross validation respectively 80, 40, 20, or 10 queries were tested, while 80, 120, 140, or 150 queries were used in training. Acoustic unit cover-

age in Table 4.7 is the averaged percentage of initial and final models appearing in the test queries that also appear in the training query set. The acoustic unit coverage in Table 4.7 is much higher than Table 4.5 as there were 4602 triphones for the lecture courses but only 147 initial plus final models for the broadcast news considered here. The experiment results showed that with more than 120 training queries significant improvements were obtained.

4.7 Summary

We presented a novel approaches for STD where acoustic model parameters are adjusted according to the results of relevance feedback. Relevance feedback with acoustic model re-estimation were shown to yield improved performance for both short- and long-term context relevance feedback. The best performance was obtained by using objective functions that take into account the nature of the retrieval task and the unlabelled segments.

Chapter 5 Machine Learning Methods with Pseudo-relevance Feedback

5.1 Introduction

There have been some previous works [8,52–55] taking advantage of the discriminative capability of machine learning methods such as support vector machines (SVM) or multi-layer perceptrons (MLP) to facilitate STD. In these works, the information from the recognition output, such as acoustic likelihood, language model scores, phone posterior probabilities, phone durations and so on, is used as features for the machine learning methods. On the other hand, techniques of using machine learning methods in relevance feedback scenario have been extensively developed for video and image retrieval, and the content of images and videos is usually directly taken as the features. The similar idea can be considered for spoken content retrieval. Instead of deriving the information from the recognition output, which may be corrupted by the poor recognition, taking the spoken content itself, that is, the acoustic vector sequences of the spoken content, as features may be more effective.

In this chapter, a new approach to improve STD using SVM is introduced [123, 124], which identifies the relevance of each segment directly from its acoustic vectors like MFCC. The concept of pseudo-relevance feedback (PRF) in Section 3.2 which was well used in the retrieval of text, image and video is considered here. PRF typically assumes that a small number of top-ranked objects in the first-pass retrieved results are relevant (or "pseudo-relevant"), and sometimes in addition some bottom-ranked objects are irrelevant

(or "pseudo-irrelevant"), and these pseudo-relevant (and -irrelevant) objects can then be taken as extra information to improve the retrieval results including used as the training data for the machine learning methods. In this way, a set of training data for each specific query can be collected for training query-specific models. Although the training data thus obtained would be noisy due to the lack of supervision, it would be quite matched to the target spoken archive. Any machine learning methods can be applied in this scenario, but since SVM yields best performance in the preliminary experiments, only the results based on SVM are reported in the following.

The approach introduced in this chapter is different from the existing works on STD in at least two ways:

1. Acoustic vector sequences such as MFCC sequences are taken as the features for discriminating relevant and irrelevant segments. This kinds of features have not been tested before.

2. The previous works trained the machine learning models from a set of external labelled data. This is the first time the scenario of PRF is successfully applied on STD with machine learning methods.

5.2 Support Vector Machines for Pseudo-relevance Feedback

Fig. 5.1 shows the framework for the proposed approach. After a query Q is entered, the spoken segments x are retrieved and ranked based on the relevance score $S(Q, x)$

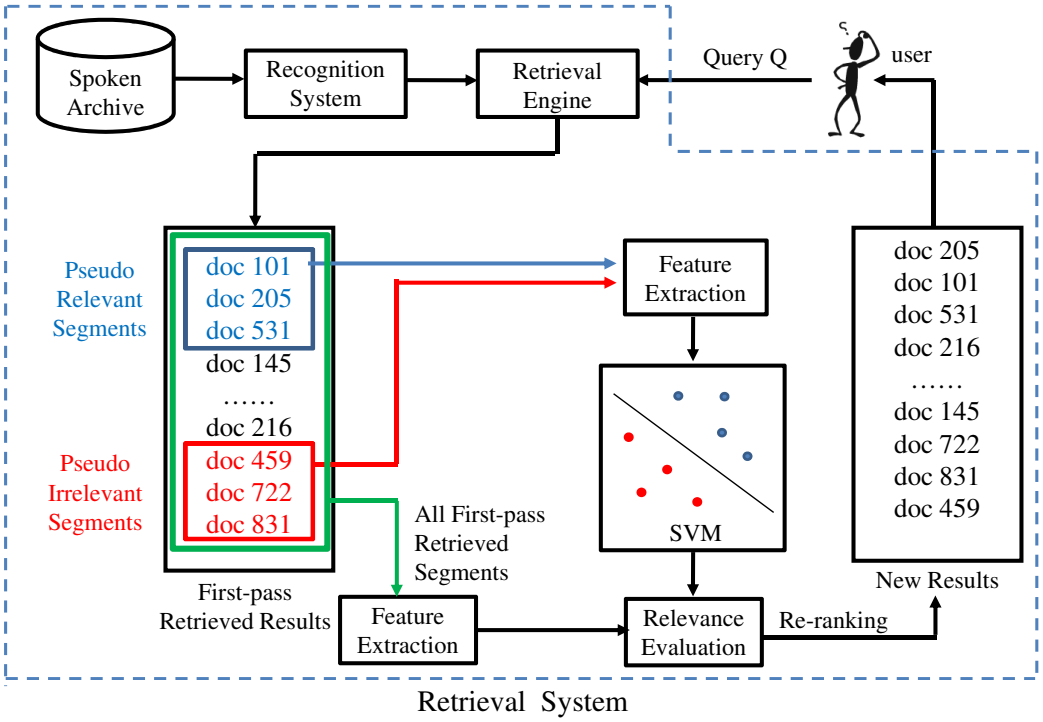


Figure 5.1: The framework for spoken term detection (STD) using support vector machines (SVM) with pseudo-relevance feedback.

in (4.1) [1]. On the left lower part of the figure is the first-pass returned list. As shown in Fig. 5.1, some spoken segments in the first-pass retrieved list are respectively taken as pseudo-relevant and -irrelevant spoken segments, and they are considered as positive and negative examples to train an SVM model, which would be used for determining the relevance between a segment and the query term Q. Based on the relevance of the spoken segments derived from the SVM model, the segments are finally re-ranked. To train such model, the acoustic vector sequence of each spoken segment x should be represented by a feature $f(x)$ as will be presented further in Section 5.3. A simple but effective way for example selection is to respectively take the top and bottom N' segments on the first-pass

[1]The notation θ is ignored in this chapter since the acoustic model parameters are not considered here.

returned list as positive and negative examples. More sophisticated approach for example selection will be described later in Section 5.4.

Suppose that the top N' segments are taken as the positive example set $\mathcal{X}_T$, while the bottom N' segments are taken as the negative example set $\mathcal{X}_F$ [2]. An SVM model represented as a weight vector w can be learned to measure the relevance of each segment with respect to the query based on the examples. The SVM model w is learned by solving the following optimization problem [125]:

$$\min_{w,\epsilon_i^t,\epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}_T} \epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}_F} \epsilon_j^f, \tag{5.1}$$

such that

$$\forall x_i^t \in \mathcal{X}_T, \quad w \cdot f(x_i^t) \geq 1 - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$

$$\forall x_j^f \in \mathcal{X}_F, \quad w \cdot f(x_j^f) \leq -1 + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

The constraints in (5.1) require that the inner products of w and the positive examples' features $f(x_i^t)$ should be larger than one, while the inner products of w and $f(x_j^f)$ smaller than negative one. Each constraint is padded with a per-example slack variable (ϵ_i^t for example x_i^t and ϵ_j^f for example x_j^f). The sum of the slack variables over the training examples is minimized to reduce the degree of constraint violations to the smallest extent. The norm of the vector w to be learned and the scale of the slack variables for positive and negative examples are respectively traded off with the parameters γ and γ'. Based on (5.1), $w \cdot f(x)$ is tend to be larger for those positive examples (or pseudo-relevant segments), and smaller for those negative examples (or pseudo-irrelevant segments), so $w \cdot f(x)$ for a retrieved segment x can measure the confidence to be relevant with respect to the query term.

[2] Since the top and bottom N' segments are selected as the examples, the sizes of $\mathcal{X}_T$ and $\mathcal{X}_F$ are equal.

SVM-derived confidence score $C_{SVM}(x)$ is then obtained by linearly normalizing $w \cdot f(x)$ into a real number between 0 and 1:

$$C_{SVM}(x) = \frac{w \cdot f(x) - d_{min}}{d_{max} - d_{min}}, \tag{5.2}$$

where d_{max} and d_{min} are respectively the maximum and minimum $w \cdot f(x)$ among all the segments in the first-pass retrieved list. The new relevance score $S'_{SVM}(Q, x)$ is then obtained by integrating the original relevance score $S(Q, x)$ in (4.1) with the confidence score $C_{SVM}(x)$ as

$$S'_{SVM}(Q, x) = S(Q, x) C_{SVM}(x)^{\delta}, \tag{5.3}$$

where δ is a weight parameter. A new ranking list is thus generated based on the new relevance scores in (5.3).

5.3 Feature Representations based on Acoustic Information

In order to train an SVM model for each query term as mentioned above, each spoken segment needs to be represented by a feature. The basic idea here is to directly use the information in the spoken content instead of recognition output. Since the MFCC vector sequences representing different occurrences of the same term should be similar in some way, while very different MFCC vector sequences very possibly imply different terms, it is therefore possible to discriminate relevant and irrelevant spoken segments by comparing the MFCC vector sequences with the pseudo-relevant and -irrelevant segments based on the hypothesized regions. In this section, we show the method representing the MFCC vector sequences as a feature.

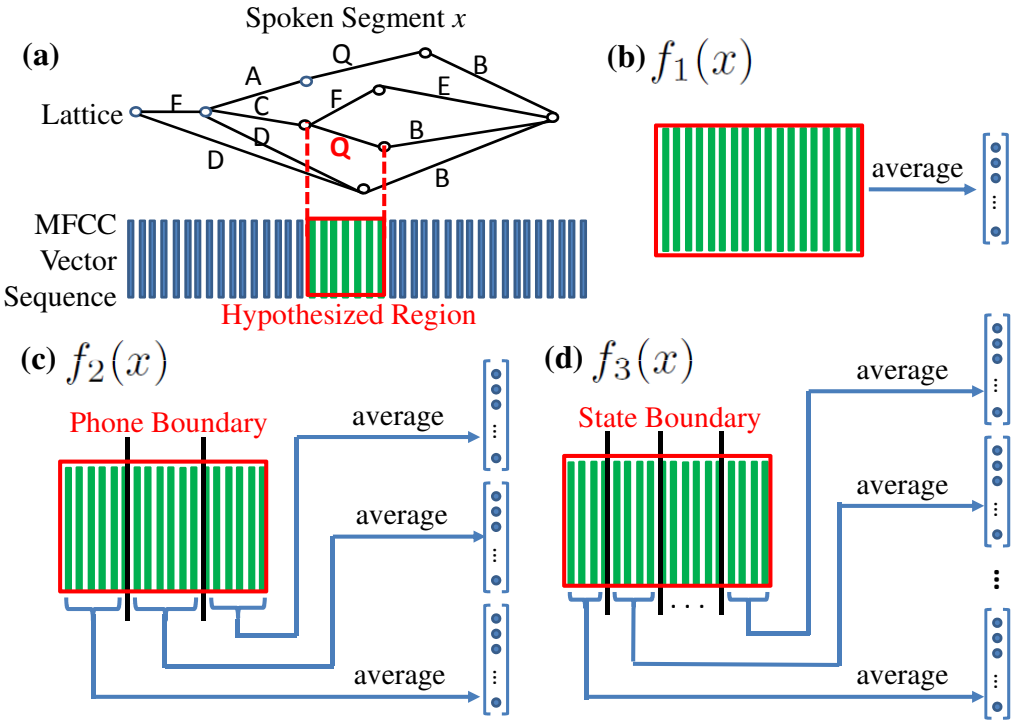


Figure 5.2: Different forms of feature representations. (a): the definition of a "hypothesized region" in the lattice of segment x for the query term Q. (b), (c) and (d): the features $f_1(x)$, $f_2(x)$ and $f_3(x)$ respectively.

Here we first define the "hypothesized region" for a spoken segment x with respect to a query Q to be the part of the MFCC vector sequence for the segment corresponding to a word arc in the lattice whose word hypothesis is exactly the query term Q with the highest posterior probability, as shown in Fig. 5.2 (a) at the upper left corner of the figure. Note that the hypothesized region is a sequence of MFCC vectors with variable length, but for model training and testing, it is more convenient to represent different spoken segments by features with fixed dimensionality. Fig. 5.2 (b), (c) and (d) illustrate three different ways to accomplish this goal as follows.

- **Term-based Average**: All MFCC vectors in the hypothesized region for the query

term are averaged into a single feature, so the dimensionality of the feature is the same as that of each MFCC vector. The value of each component of the feature is the average of all of the corresponding components of the MFCC vectors (or the corresponding MFCC parameters) in the hypothesized region. This is denoted by $f_1(x)$ and is shown in Fig. 5.2 (b) at the upper right corner of the figure.

- **Phone-based Average and Concatenation**: The hypothesized region is segmented into a sequence of phone segments based on the phone boundaries obtained during the lattice construction. Each phone segment is then represented by the average of the MFCC vectors in the phone segment. The concatenation of these averaged MFCC vectors representing the phone segments then gives the feature for a spoken segment. Thus for a query term including m phones the dimensionality of the feature is m times of the dimensionality of a single MFCC vector. This is denoted by $f_2(x)$ and shown in Fig. 5.2 (c) at the lower left corner of the figure.

- **State-based Average and Concatenation**: Each phone segment is further segmented into a sequence of state segments according to the HMM state boundaries obtained during the recognition, each of which is again represented by the average of the MFCC vectors. All these averaged vectors for HMM states in a hypothesized region are then concatenated as a feature. Thus for l-state phone HMMs the dimensionality of such a feature is l times of the dimensionality of $f_2(x)$. This is denoted as $f_3(x)$ and is shown in Fig. 5.2 (d) at the lower right corner of the figure.

Although in the above we only mention MFCC vectors, and in the experiments below only results using MFCC vectors are reported, it is clear that many other representations for acoustic information of speech can be used. A good example may be the Gaussian

posteriorgrams [40,126] which may take better care of the speaker variability issue since the target spoken segments may be produced by many different speakers.

5.4 Enhanced Pseudo-relevance Feedback

In conventional PRF scenario, some top/bottom-ranked segments in the first-pass results are usually taken as positive/negative examples. However, it is unavoidable to include some incorrect examples (irrelevant segments are taken as positive examples, and vice versa) in the training data especially when the quality of the recognition output is relatively poor. To better handle this problem, a set of examples not restricted to the top and bottom segments is carefully selected, and the reliability for each selected example is further estimated. The formulation of SVM is modified to make the machine only focus on the presumably correct examples during training, and compel the unreliable examples to have little influence upon the model learned.

5.4.1 Example Selection and Reliability Estimation based on Acoustic Similarity

Because the top/bottom-ranked segments in the first-pass results usually have large probabilities to be relevant/irrelevant, the relevance of each segment x can be estimated to some extent based on the similarity between the feature of x and the features of the top/bottom segments. According to the above principle, we can obtain an example set not restricted to top and bottom segments. For each segment in the first-pass retrieved list, its similarity with the top and bottom segments is first computed based on the distances between

their features. If a spoken segment x is similar to more top-ranked segments than bottom-ranked segments, it would be taken as positive examples, and the difference between its similarity to top- and bottom-ranked segments can be taken as the reliability of the example. On the contrary, a spoken segment similar to more bottom-ranked segments is a negative example, and its reliability can be evaluated in the same way.

Based on the above statement, the following procedure is derived to obtain a set of positive examples $\mathcal{X}'_T$ and negative examples $\mathcal{X}'_F$ in which each example x has a value $C(x)$ representing its reliability.

1. Each segment x in the first-pass result is first assigned an initial score $w_0(x)$, which is 1 for top N' segments, -1 for bottom N' segments, and 0 for the others.

2. Compute the similarity $s(x_i, x_j)$ between any two segments x_i and x_j in the first-pass results based on the Euclidean distance of their features,

$$s(x_i, x_j) = exp(-\frac{||f(x_i) - f(x_j)||_2}{\sigma}), \tag{5.4}$$

   where $f(x_i)$ is the feature of segment x_i, which can be either $f_1(x_i)$, $f_2(x_i)$ or $f_3(x_i)$ in Section 5.3, and σ is the variance of $||f(x_i) - f(x_j)||_2$ for all segment pairs. Smaller $||f(x_i) - f(x_j)||_2$ implies larger $s(x_i, x_j)$.

3. Find the K nearest neighbours for each segment x_i based on $s(x_i, x_j)$, which is denoted as $N(x_i)$.

4. Then a score $w(x_i)$ is computed for each segment x_i, which would be used in the next step for example selection. $w(x_i)$ is the interpolation of x_i's initial score $w_0(x_i)$ obtained in step (1) and its mutual nearest neighbours' initial scores $w_0(x_j)$ weighed

by their similarities $s(x_i, x_j)$:

$$w(x_i) = (1-\alpha)w_0(x_i) + \alpha \sum_{\substack{x_j, \\ x_j \in N(x_i), \\ x_i \in N(x_j)}} s(x_i, x_j) w_0(x_j), \tag{5.5}$$

where x_j is a K mutual nearest neighbour of x_i [3], and α is the interpolation weight. Based on (5.5), if most of x_i's neighbours are top N' segments, $w(x_i)$ would be positive with large value; likewise, if most of x_i's neighbours are bottom N' segments, $w(x_i)$ would be very negative.

5. The segments x with positive $w(x)$ in (5.5) are taken as positive examples for SVM training, while the segments with negative $w(x)$ are negative examples. The **absolute value** of $w(x)$ is regarded as the reliability for example x, which is denoted as $C(x)$.

If α in (5.5) is 0, the above procedure reduces to taking top N' and bottom N' segments as training examples, and the reliability $C(x)$ for each example would be 1.

5.4.2 Modified Support Vector Machines

From the procedure in the last subsection, a set of positive examples $\mathcal{X}'_T$ and a set of negative examples $\mathcal{X}'_F$ are obtained. For each example x in $\mathcal{X}'_T$ and $\mathcal{X}'_F$, there is a non-negative real number $C(x)$ representing the example's reliability. To compel the unreliable examples to have less influence upon the model learned, the formulation of SVM in (5.1) is modified. There are three possible modifications:

- *Slack Variables Rescaling* [127]: In this approach, the slack variable corresponding to each example x is multiplied by the reliability $C(x)$, and the formulation for

[3] x_j is x_i's K nearest neighbours, while x_i is also x_j's K nearest neighbours.

SVM is modified as

$$\min_{w,\epsilon_i^t,\epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}'_T} C(x_i^t)\epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}'_F} C(x_j^f)\epsilon_j^f, \tag{5.6}$$

such that

$$\forall x_i^t \in \mathcal{X}'_T, \quad w \cdot f(x_i^t) \geq 1 - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$

$$\forall x_j^f \in \mathcal{X}'_F, \quad w \cdot f(x_j^f) \leq -1 + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

In (5.6), because the slack variables for the examples with larger reliabilities are multiplied by larger $C(x)$, to minimize (5.6), obeying the constraints for the examples with larger reliabilities would be given precedence over other constraints. Therefore, the weight vector w learned from (5.6) would be less dependent on the examples with small reliabilities.

- *Margins Rescaling*: Here the reliability $C(x)$ is regarded as the margin of the constraint for the example x, so the SVM formulation is thus modified:

$$\min_{w,\epsilon_i^t,\epsilon_j^f} \frac{1}{2}\|w\|_2 + \gamma \sum_{x_i^t \in \mathcal{X}'_T} \epsilon_i^t + \gamma' \sum_{x_j^f \in \mathcal{X}'_F} \epsilon_j^f, \tag{5.7}$$

such that

$$\forall x_i^t \in \mathcal{X}'_T, \quad w \cdot f(x_i^t) \geq C(x_i^t) - \epsilon_i^t, \quad \epsilon_i^t \geq 0$$

$$\forall x_j^f \in \mathcal{X}'_F, \quad w \cdot f(x_j^f) \leq -C(x_j^f) + \epsilon_j^f, \quad \epsilon_j^f \geq 0.$$

In (5.7), the examples with larger reliabilities would be equipped with larger margins. Hence, to minimize (5.7), a weight vector w would be learned to give reliable positive examples larger $w \cdot f(x)$, while the suspected ones smaller $w \cdot f(x)$, and vice versa for the negative examples. The model w thus learned can well separate

positive and negative examples with large reliabilities due to their larger margins, and pays less attention on discriminating the unreliable ones.

- *Slack Variables & Margins Rescaling*: Certainly, it is possible to resale the slack variables and margins at the same time.

When the reliability $C(x)$ equals 1 for all the training examples, all the above modified SVM are reduced to ordinary SVM in (5.1).

5.5 Experiments for Lecture Courses

5.5.1 Experimental Setup

Mean Average Precision (MAP) was used as the retrieval performance evaluation measure. Pair-wise t-test with significance level at 0.05 was used to test the significance for the performance improvement. The package, CVXOPT [4], was used for solving the SVM optimization problems. γ and γ' in (5.1), (5.6) and (5.7) were set to be the inverse of the average of the training features' norms [5].

The lecture courses used in Section 4.5 were also tested here. 162 manually selected Chinese queries were tested here, each consisting of a single word. We used four sets of acoustic models for generating the lattices (three of them were also used in Section 4.5):

- Speaker Independent Model (SI): As described in Section 4.5.1.
- Speaker Adaptation Model 1 (ADP1): As above.
- Speaker Adaptation Model 2 (ADP2): As above.

[4] http://abel.ee.ucla.edu/cvxopt/

[5] SVM-light uses the same strategy to derive the parameters.

- Speaker Dependent Model (SD): trained on the 12-hour data which came from the course of the same instructor but different from the testing archive here with 6620 state-tied triphones spanned from 35 Mandarin monophones and 39 English monophones. The models included triphones developed from the phoneme set including both Mandarin and English phonemes, so it was possible to transcribe the English words correctly. The character accuracies (for Chinese parts only) of the 1-best transcriptions for the models were 84.08%.

5.5.2 Features based on Acoustic Information

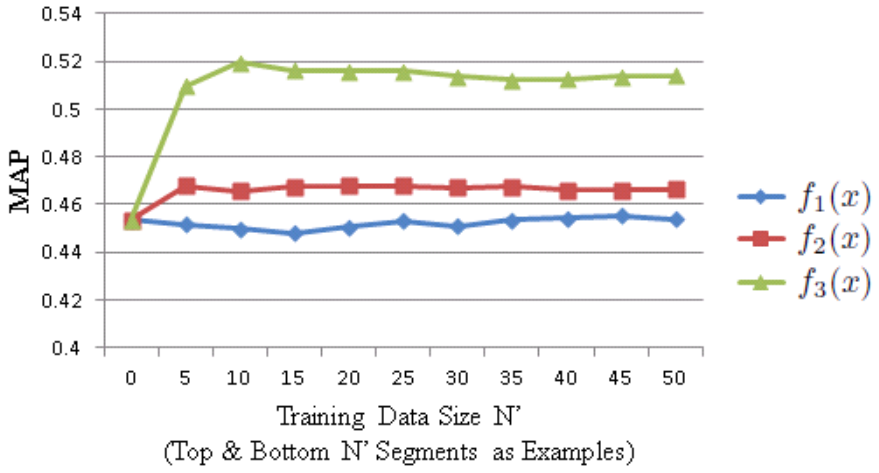


Figure 5.3: MAP performance yielded with features $f_1(x)$, $f_2(x)$ and $f_3(x)$ in Section 5.3 when top/bottom N' segments in the first-pass results were selected as positive/negative examples. The speaker independent (SI) models were used in the experiments.

First of all, we tested the performance of features $f_1(x)$, $f_2(x)$ and $f_3(x)$ in Section 5.3 when top and bottom N' segments in the first-pass results ($\mathcal{X}_T$ and $\mathcal{X}_F$) were selected as examples to train the SVM model. Fig. 5.3 shows the MAP performance yielded with features $f_1(x)$, $f_2(x)$ and $f_3(x)$ in Section 5.3 as functions of N', or number

of top/bottom segments taken as examples. Here N' was set from 5 to 50 with intervals of 5 [6]. The speaker independent (SI) models were used in the experiments of Fig. 5.3. The points for $N' = 0$ represent the original first-pass results which are taken as the baselines. We found that $f_1(x)$ yielded no improvement since the query term usually included a sequence of phonemes, but the acoustic characteristics of the different phonemes are averaged and smoothed in $f_1(x)$, which is too coarse to represent the hypothesized region. More sophisticated feature representations, $f_2(x)$ or $f_3(x)$, yielded improvements because the acoustic characteristics for each phoneme or even each HMM state were represented in these features, which better represented the hypothesized region. $f_3(x)$ obviously performed the best, which implied the HMM states were able to represent the acoustic characteristics within a hypothesized region.

Table 5.1 shows the MAP performance yielded with the feature $f_3(x)$ when different numbers of top/bottom segments were considered as examples. N' in Table 5.1 is the number of top/bottom segments taken as training examples. The four columns are respectively the results for four different sets of acoustic models, SI, ADP1, ADP2 and SD. The first-pass results obtained before PRF are taken as the baselines, and the superscript labels * indicate significantly better than the baselines. The SVM trained with feature $f_3(x)$ when taking top and bottom N' as examples always offered some improvements as compared to the baselines no matter the acoustic models used for generating the lattices. The improvements achieved were always significant regardless of N' for SI, ADP1 and ADP2 models. From Table 5.1, we also observed that as the example size N' was raised the MAP first increased and then slightly decreased in most cases. This is reasonable because larger

[6] If in the first-pass results there were fewer than $2N'$ spoken segments, N' was simply set to half of the number of retrieved segments.

N' implied more training data were used in training the SVM model, and the disturbances caused by the incorrect assumption about the relevance of the training segments (irrelevant segments assumed to be relevant and vice versa) can be diluted. However, when N' was too large, since there were only limited number of relevant segments for each query, some irrelevant segments were inevitably included in the pseudo-relevant training set and taken as relevant, which caused the degradation for PRF.

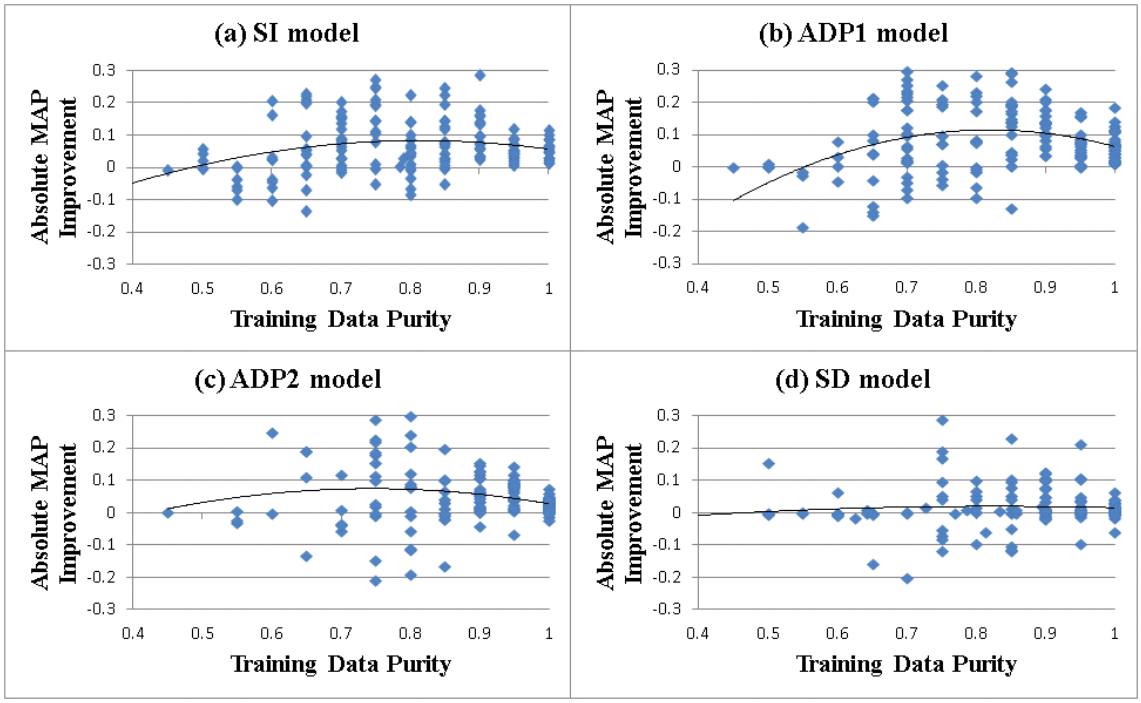


Figure 5.4: Distribution of absolute MAP improvement versus the training data purity for SVM training for each query with feature $f_3(x)$ when taking top and bottom 10 segments as examples ($N' = 10$ in Table 5.1). (a), (b), (c) and (d) are respectively for the results with different sets of acoustic models, SI, ADP1, ADP2 and SD. Training data purity is the average of the percentages of pseudo-relevant segments being relevant and pseudo-irrelevant segments being irrelevant. Each point in the figures represents a query. The curves in the figures are the quadratic trend lines.

Since we can not ensure all the pseudo training data is correct, PRF is not able to improve the performance of every query. Usually PRF improves the performance of some queries but hurts the others. We are interested to see how SVM with $f_3(x)$ performed with such corrupted training data. We first define the purity of the training data for each query as the average of the percentages of pseudo-relevant segments being actually relevant and pseudo-irrelevant segments being actually irrelevant. Fig. 5.4 shows the distribution of the absolute MAP improvement achieved with each query versus the purity of the training data for that query with $f_3(x)$ when taking top and bottom 10 segments as examples ($N' = 10$ in Table 5.1). Fig. 5.4 (a), (b), (c) and (d) are respectively for four different sets of acoustic models, SI, ADP1, ADP2 and SD. Each point in the figures represents one query, with vertical scales being the absolute MAP improvement for the query, and the horizontal scales being the purity of training data. Negative improvement means the MAP performance for the query was actually degraded after PRF. The curves in the figures are the quadratic trend lines. At the first glance, it seems surprising that higher training data purity did not always imply larger MAP improvement. This is probably because the query with higher training data purity usually has higher MAP for the first-pass retrieved results, the space left for further improvement is therefore limited. Although the very corrupted training data really degraded the performance, we observed that even though the training data purity was less than 70%, the improvements could still be achieved for some queries.

Table 5.2 shows the percentage of queries degraded after PRF with feature $f_3(x)$ when taking top and bottom 10 segments as examples ($N' = 10$ in Table 5.1). The four columns correspond to the results with four different acoustic models, SI, ADP1, ADP2, and SD. The results with ADP2 model achieved the lowest degradation rate which is less

than 15%, or the performance of more than 85% queries can be improved after PRF.

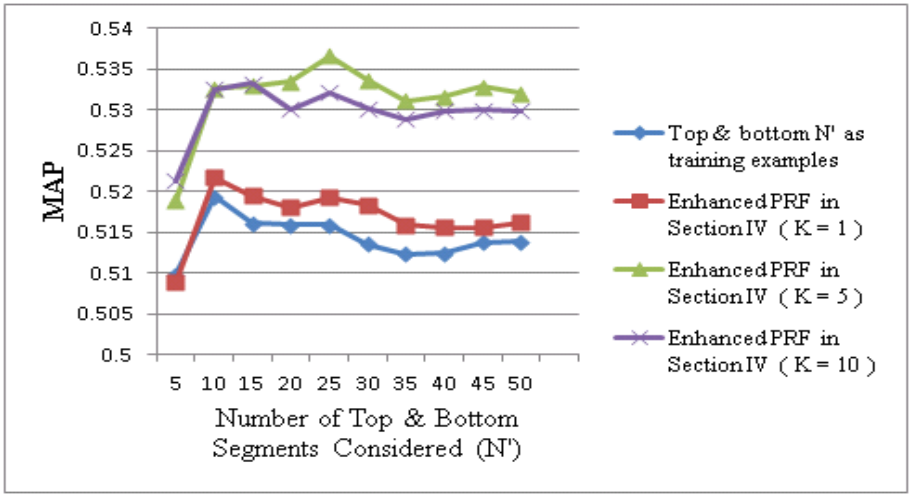
5.5.3 Enhanced Pseudo-relevance Feedback

In this section, the enhanced PRF described in Section 5.4 was tested and analysed. Because the feature $f_3(x)$ yielded the best results in Subsection 5.5.2, it was used for representing a spoken segment in the following experiments.

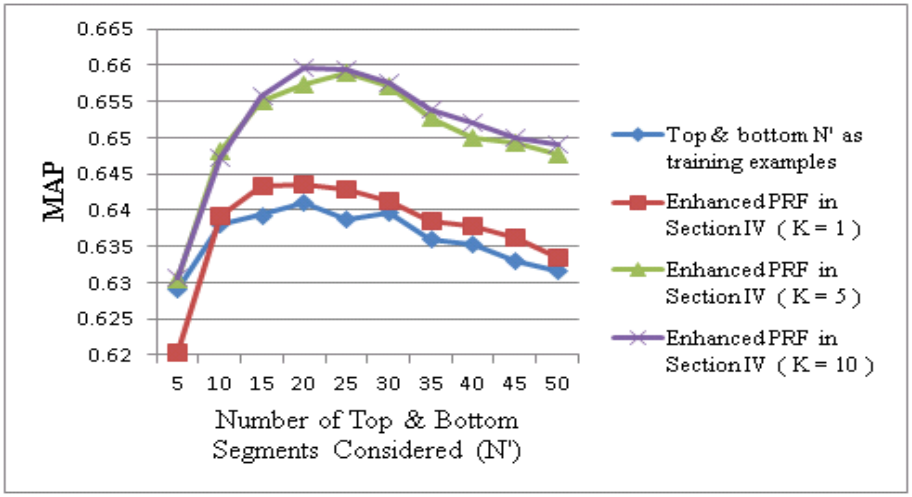
In Table 5.3, MAP performance yielded by enhanced PRF in Section 5.4 is presented. The SI model was used in the experiments. Column (a) is the results taking top and bottom N' segments ($\mathcal{X}_T$ and $\mathcal{X}_F$) as training examples. The results in Column (a) have been reported in the SI column of Table 5.1. Section (b) is for enhanced PRF ($\mathcal{X}'_T$ and $\mathcal{X}'_F$ as training example sets). The variable N' for enhanced PRF denotes the number of top and bottom segments assigned non-zero initial scores $w_0(x)$ at the step (1) of the procedure in Subsection 5.4.1. α in (5.5) was 0.8, and K at the step (3) of the procedure in Subsection 5.4.1 was 5. $f_3(x)$ was used for both SVM training and computing the similarity in (5.4). As described in Subsection 5.4.2, the example reliabilities can be considered in SVM training by three methods, *Slack Variables Rescaling*, *Margins Rescaling*, and *Slack Variables & Margins Rescaling*, each corresponds to a column in Section (b). The superscript labels † indicate significantly better than the results in column (a) under equal N'. We observed that the results based on *Slack Variables Rescaling* (column (b-1)) could not surpass the baselines (column (a)), whereas *Margins Rescaling* (column (b-2)) outperformed the results of taking top and bottom segments as examples (column (a)) regardless of N'. This is probably because *Margins Rescaling* in (5.7) utilized the reliabilities in a more aggressive way than *Slack Variables Rescaling* in (5.6). When utilizing *Slack Vari-*

ables Rescaling, $C(x)$ in (5.6) was malfunctioned as the constraint corresponding to the example x was not violated, so the reliabilities $C(x)$ for most examples would not have any effect on the training results. On the other hand, since *Margins Rescaling* considered the reliabilities as the margins of the constraints, every example's reliability would have some influences upon the model learned. Certainly *Slack Variables & Margins Rescaling* (column (b-3)) offered the greatest improvements, and the improvements over the baselines (column (a)) were significant regardless of N'. In the following experiments, *Slack Variables & Margins Rescaling* was always used for considering the example reliabilities $C(x)$.

Fig. 5.5 shows MAP performance yielded by enhanced PRF in Section 5.4 as functions of N' with feature $f_3(x)$. Note that the variables N' at the horizontal scales are the numbers of top and bottom segments used for SVM training for the baselines; or the numbers of top and bottom segments assigned non-zero initial scores for enhanced PRF at the step (1) of the procedure in Subsection 5.4.1. Fig. 5.5 (a), (b), (c) and (d) are respectively for different sets of acoustic models, SI, ADP1, ADP2 and SD. Taking top and bottom segments as training examples is the blue lines (with a rhombus) in the figures, and the other lines are for enhanced PRF with different K, which is the number of nearest neighbours at the step (3) of the procedure in Subsection 5.4.1. α in (5.5) was 0.8. We observed that enhanced PRF in Section 5.4 obtained improvements with SI, ADP1 and ADP2 models regardless of N', except $N' = 5$. This shows the effectiveness of the proposed approach. Remarkable improvements over the baselines were not observed with the SD models. Since the SD models had extremely high quality which enabled the first-pass results to be ranked almost perfectly, in that case selecting top and bottom

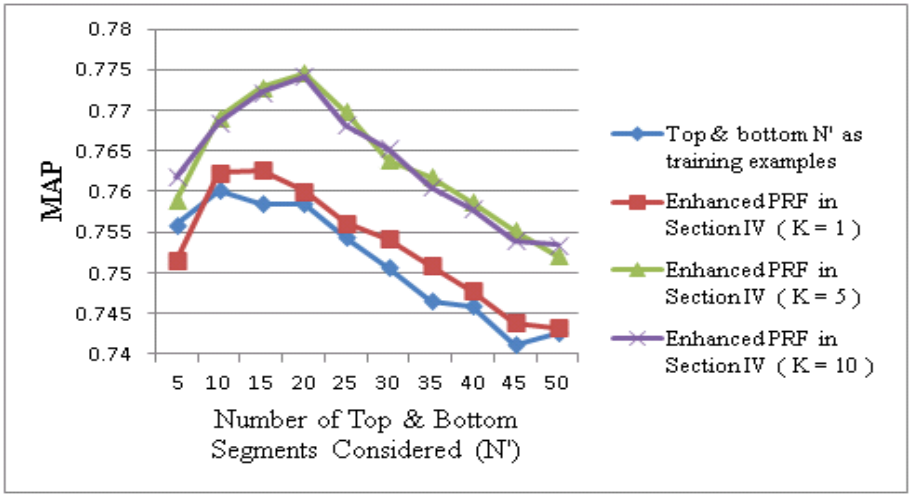


(a) SI model

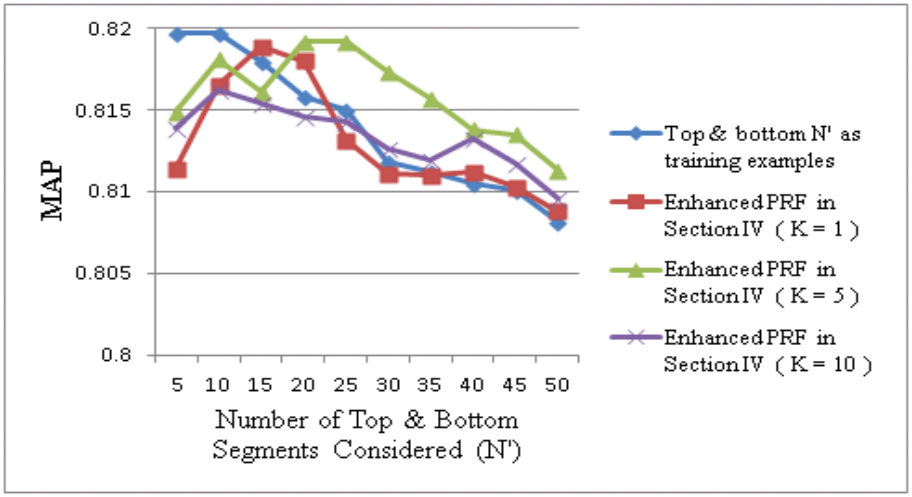


(b) ADP1 model

Figure 5.5: MAP performance yielded by enhanced PRF in Section 5.4 as functions of N' with feature $f_3(x)$. N' is the number of top and bottom segments considered. K is the number of nearest neighbours at the step (3) of the procedure in Subsection 5.4.1.



(c) ADP2 model



(d) SD model

Figure 5.5: MAP performance yielded by enhanced PRF in Section 5.4 as functions of N' with feature $f_3(x)$. N' is the number of top and bottom segments considered. K is the number of nearest neighbours at the step (3) of the procedure in Subsection 5.4.1.

N' segments may be sufficient to generate an example set with very high training data purity. In such condition, the proposed example selection approach and reliability estimation method may not offer too much benefit. For the enhanced PRF with SI model, $K = 5$ had slightly better performance than $K = 10$, and the performance of $K = 5$ and $K = 10$ was comparable when ADP1 and ADP2 models were used. This implies that enhanced PRF was not very sensitive to the value of K as long as K was large enough to consider sufficient neighbours. When K was equal to 1, subtle improvements over the baselines were still observed.

Table 5.4 presents the MAP performance yielded by enhanced PRF in Section 5.4 with different α in (5.5). K and N' in the procedure of Subsection 5.4.1 were fixed to be 5 and 10 respectively. The four columns correspond to the results with four different acoustic models, SI, ADP1, ADP2, and SD. The results with $\alpha = 0.8$ have been reported in Fig 5.5. $\alpha = 0.0$ represents simply taking top and bottom segments as training examples. The superscript labels † indicate significantly better than the baselines. The greatest results in each column were in bold. It is observed that with SI, ADP1 and ADP2 models when the values of α were raised, the improvements increased accordingly. The peaks of the improvements were achieved when α was 0.9, 0.8 and 0.9 for SI, ADP1 and ADP2 respectively. This implies that in those conditions the reliabilities estimated by the neighbours were quite useful, so this factor should be weighted more when estimating the reliabilities. As we have observed in Fig 5.5 (d), the example selection method did not provide too much benefit for the SD model, so smaller α was desired in that case.

5.6 Experiments for Broadcast News

Here we also tested the proposed approaches on the broadcast news corpus. The experimental setup here was exactly the same as that in Section 4.6.1.

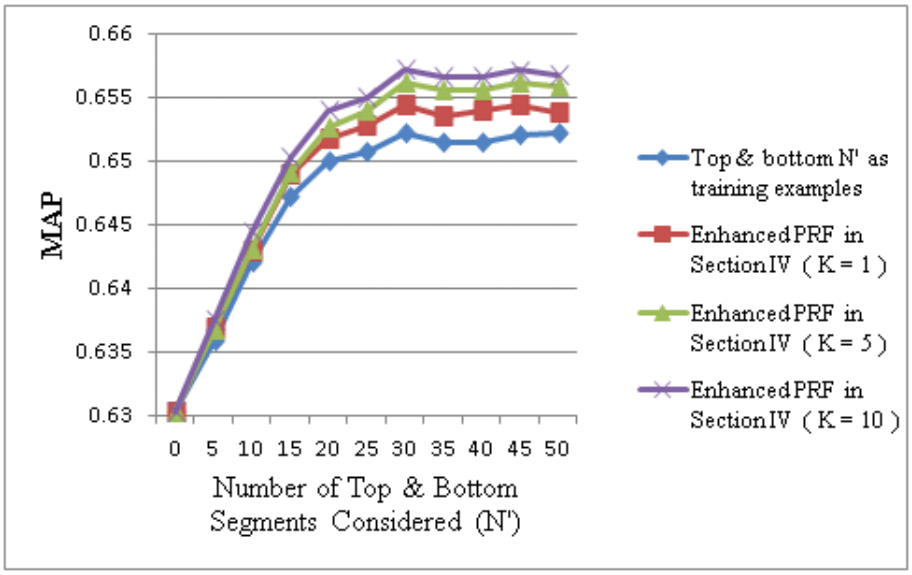


Figure 5.6: MAP performance for broadcast news yielded by PRF with SVM in Section 5.4 as functions of N' with feature $f_3(x)$ described in Section 5.3. N' at the horizontal scales is the number of top and bottom segments considered. $N' = 0$ represents the baselines without PRF. Taking top and bottom segments as examples is the blue line (with rhombuses) in the figure, and the other lines in the figure are for enhanced PRF. K is the number of nearest neighbours at the step (3) of the procedure in Subsection 5.4.1.

Fig. 5.6 is the MAP performance for the broadcast news corpus yielded by PRF with SVM with feature $f_3(x)$ described in Section 5.3. N' at the horizontal scales is the number of top and bottom segments considered. $N' = 0$ represents the baselines without PRF. Taking top and bottom segments as examples is the blue line (with rhombuses) in the figure, and the other lines in the figure are for enhanced PRF. K is the number of nearest neighbours at the step (3) of the procedure in Subsection 5.4.1. The results observed

on the broadcast news were consistent with those on the lecture courses. For the results of taking top and bottom N' spoken segments in the first pass as training data (the blue line in Fig. 5.6), the improvements yielded with feature $f_3(x)$ were *significant* expect $N' = 5$. Because the broadcast news contains speech covering many different speakers and environments, more training data was necessary to cover enough acoustic variations. The enhanced SVM was clearly superior than simply selecting top and bottom segments regardless of N' and K.

5.7 Experiences

Some unsuccessful results not reported are briefly summarized in this section. Besides representing each spoken segment based on acoustic vector sequences such as MFCC, there are certainly lots of possible alternatives in the literatures like acoustic likelihood, language model scores, context of the query hypotheses, positions of the hypotheses, phone duration and so on. Among all of the above alternatives, phone duration led to the best results. Some previous researches [55] also pointed out that phone duration is a useful feature for spoken term detection because the hypotheses with extremely short phone durations usually imply insertion errors. However, the feature representations proposed in Section 5.3 were much better than phone or state durations in terms of performance in the experiments.

Any machine learning method can be used to replace SVM in the scenario of Section 5.2 here. However, among all the methods tested, SVM obtained the best results. Adaboost resulted in very bad results (even worse than the baselines without PRF). Since Adaboost focuses on the data points hard to be correctly classified by weighting those

data points more at each training iteration, it may over fit to the incorrect training data. Since Adaboost is susceptible to the noisy data [128], it is not suitable for PRF. Due to the very limited training data, ordinary MLP did not result in good results, but with proper regularization, MLP was comparable to SVM.

The learning-to-rank techniques like Ranking SVM [82] or SVM-MAP [129] which maximize some criteria related to the retrieval evaluation measures can be considered here. However, since the training data obtained by PRF is usually noisy, optimizing the retrieval evaluation based on these noisy training examples may not be helpful. Thus in the experiments ranking SVM can not outperform ordinary SVM. One-class SVM [130] which neglects the negative examples was also tested, but it could not outperform ordinary SVM either.

Moreover, since there are limited data points with relatively high dimensions in the task considered, dimension reduction may be helpful. However, only PCA yielded very subtle improvements. Other methods taking the labels of the training data into consideration like LDA did not result in any improvements since the training data is noisy. Biased Discriminant Analysis [131] which was proposed for image retrieval did not offer improvements either. It is also possible to estimate a transformation separating the features of the segments with distant relevance scores in the first pass [132], but no improvement was observed by this approach.

Relevance feedback is a very suitable field for testing semi-supervised machine learning methods [133]. The spoken segments with relevance information from either user relevance feedback or pseudo-relevance feedback can be considered as labelled data, and the other spoken segments retrieved in the first pass are unlabelled data. It is intuitive

that these unlabelled data can enhance the training on the labelled data. However neither Transductive SVM [134] nor Laplacian SVM [135] outperformed the ordinary SVM in the experiments.

5.8 Summary

In this chapter, pseudo-relevance feedback is used to automatically generate training data for query-specific SVM, and then the SVM further re-ranks the first-pass retrieved results. The features based on acoustic information were defined and used in training the SVM, and the enhanced PRF with better example selection strategy, example reliability estimation, and modified SVM was introduced. The proposed approaches were tested under different recognition accuracies, and significant improvements were obtained in most cases.

Table 5.1: MAP performance yielded with feature $f_3(x)$ in Section 5.3 when different numbers of top/bottom segments in the first-pass results were selected as positive/negative examples. The four columns correspond to the results with four different acoustic models, SI, ADP1, ADP2, and SD. The first-pass results obtained before PRF are taken as the baselines, and the superscript labels * indicate significantly better than the baselines.

		SI	ADP1	ADP2	SD
first pass (baseline)		0.4536	0.5539	0.7111	0.8041
Training Data Size (N')	$N' = 5$	0.5098*	0.6290*	0.7559*	0.8197*
	$N' = 10$	0.5194*	0.6381*	0.7602*	0.8197*
	$N' = 15$	0.5161*	0.6393*	0.7584*	0.8179*
	$N' = 20$	0.5159*	0.6410*	0.7585*	0.8158*
	$N' = 25$	0.5159*	0.6387*	0.7544*	0.8150*
	$N' = 30$	0.5136*	0.6397*	0.7506*	0.8118
	$N' = 35$	0.5123*	0.6359*	0.7465*	0.8112
	$N' = 40$	0.5124*	0.6352*	0.7459*	0.8105
	$N' = 45$	0.5138*	0.6330*	0.7411*	0.8101
	$N' = 50$	0.5139*	0.6315*	0.7425*	0.8081

Table 5.2: The percentage of queries degraded after PRF with feature $f_3(x)$ when taking top and bottom 10 segments as examples ($N' = 10$ in Table 5.1). The four columns correspond to the results with four different acoustic models, SI, ADP1, ADP2, and SD.

	SI	ADP1	ADP2	SD
Percentage of Queries Degraded (%)	16.05	14.81	14.20	27.78

Table 5.3: MAP performance yielded by enhanced PRF in Section 5.4 with feature $f_3(x)$ and SI models. Column (a) is the results taking top and bottom segments as training examples, which are taken as the baselines here. Section (b) is for enhanced PRF. The example reliabilities is considered in SVM training by three methods, *Slack Variables Rescaling*, *Margins Rescaling*, and *Slack Variables & Margins Rescaling*, each corresponds to a column in Section (b). The superscript labels † indicate significantly better than the results in column (a).

Number of top & bottom segments considered (N')	(a) top & bottom segments as training examples	(b) Enhanced PRF in Section 5.4		
		(b-1) *Slack Variables Rescaling*	(b-2) *Margins Rescaling*	(b-3) *Slack Variables & Margins Rescaling*
5	0.5098	0.4979	0.5206†	0.5189†
10	0.5194	0.5088	0.5248	0.5327†
15	0.5161	0.5112	0.5189	0.5330†
20	0.5159	0.5115	0.5160	0.5335†
25	0.5159	0.5145	0.5178	0.5366†
30	0.5136	0.5143	0.5202	0.5337†
35	0.5123	0.5126	0.5175	0.5311†
40	0.5124	0.5142	0.5185	0.5316†
45	0.5138	0.5153	0.5200	0.5329†
50	0.5139	0.5156	0.5191	0.5320†

Table 5.4: MAP performance yielded by enhanced PRF in Section 5.4 with different α in (5.5). K and N' in the procedure in Subsection 5.4.1 were fixed to be 5 and 10 respectively. The four columns correspond to the results with four different acoustic models, SI, ADP1, ADP2, and SD. The superscript labels † indicate significantly better than the baselines. The greatest results in each column were in bold.

	SI	ADP1	ADP2	SD
Top & bottom N' as training examples ($\alpha = 0.0$)	0.5194	0.6381	0.7602	0.8197
$\alpha = 0.1$	0.5223†	0.6390	0.7616	0.8206
$\alpha = 0.2$	0.5239†	0.6397	0.7635	0.8213
$\alpha = 0.3$	0.5256†	0.6408†	0.7642†	**0.8217**
$\alpha = 0.4$	0.5268†	0.6418†	0.7646	0.8214
$\alpha = 0.5$	0.5289†	0.6432†	0.7656†	0.8215
$\alpha = 0.6$	0.5309†	0.6456†	0.7672†	0.8211
$\alpha = 0.7$	0.5318†	0.6472†	0.7692†	0.8194
$\alpha = 0.8$	0.5327†	**0.6481**†	0.7691†	0.8181
$\alpha = 0.9$	**0.5359**†	0.6459†	**0.7712**†	0.8145

Chapter 6 Example-based Approaches

6.1 Introduction

As mentioned, in most approaches of spoken term detection (STD), the spoken utterances are first recognized and transformed into transcriptions or lattices by speech recognition technologies, and then the search engine looks through all the transcriptions or lattices very similar to the text-based information retrieval. The recognition process can be considered as "quantization", in which the acoustic vector sequences are quantized into word symbols. Because different vector sequences may be quantized into the same symbol, much of the information in the spoken content may be lost in the stage of speech recognition, especially when the acoustic models used are not well matched to the characteristics of the acoustic signals, which naturally results in degraded recognition accuracy and poor detection performance. This is very common in the scenario of spoken content retrieval, because the huge quantities of spoken content available over the Internet are naturally produced by many different people under many different acoustic conditions, it is thus very difficult to train a set of acoustic models well matched to so many different acoustic conditions. As a result, when the relevance scores such as the posterior probabilities of the query term derived from transcriptions or lattices are used to rank the retrieved spoken segments, it is hard to judge whether a word hypothesis of the query in the transcriptions or lattices is a positive target or a false alarm when the recognition output is unreliable. Therefore, information straightly from the acoustic vector space is considered in this chapter to compensate for the recognition output.

Consider Fig 6.1, in which each point represents a spoken segment. To simplify

the description, in this example it is assumed that each spoken segment only contains an isolated word, although what would be actually considered in the following experiments is more complicated. The distance in the space of Fig 6.1 is the distance between the acoustic vector sequences of the spoken segments, which can be, for example, Dynamic Time Warping (DTW) distance. The triangles represent the relevant segments containing query Q, while the crosses represent the irrelevant segments containing word W. Because of the mismatch between the acoustic model and the target spoken archive, some spoken segments for word W are incorrectly recognized into word Q. The spoken segments in the blue circle (or the larger circle) are the ones recognized into query Q. When the query Q is entered, a text-based retrieval system may retrieve all the spoken segments in the blue circle including the ones being irrelevant actually (those crosses in the blue circle). However, since a given word may be pronounced in a similar way and thus exhibit similar acoustic vector sequences, as shown in Fig 6.1 those relevant segments may be close to each other in acoustic vector space. This acoustic vector similarity between the spoken segments is useful for STD.

One way to apply the acoustic vector similarity is based on the PRF [38,118,119, 136] scenario in Section 3.2. In this approach, given a user query, the retrieval engine first searches through the lattices to produce a first-pass returned list ranked according to a relevance score derived from the lattices. The returned segments with the highest relevance scores (most confident to be relevant) are then defined as the pseudo-relevant set. The similarities between each first-pass retrieved spoken segment and the pseudo-relevant set are computed based on the acoustic vectors of their query hypotheses, and the first-pass returned list is re-ranked accordingly. If the first-pass retrieval results are

good enough, and the relevant segments dominate the pseudo-relevant set, the re-ranking would improve the performance. Based on the example in Fig 6.1, suppose the segments in the green circle (or the smaller circle) are taken as pseudo-relevant set, the system can then identify the triangles at the lower right corner which are closer to the pseudo-relevant set are relevant, whereas the crosses at the upper left corner are irrelevant. In addition, selecting some segments with lowest relevant scores as pseudo-irrelevant set may also be helpful.

On the other hand, as described in Section 2.4, many query-by-example techniques have been developed recently. These approaches let the user use some audio examples as spoken queries to find more similar spoken content. However, sometimes it may be troublesome for a user to find some audio examples for the spoken word he/she wants to retrieve [1]. The approaches described in this chapter are very good applications for those query-by-example techniques. The example-based PRF can be regarded as selecting some audio examples in an automatic way, and use the audio examples to enhance the retrieval performance. Although only the simple DTW-based approach is applied in the proposed method, any state-of-the-art query-by-example techniques can be applied in this approach.

The PRF approach can be taken one step further with graph-based re-ranking [137–139]. In this approach, we construct for the first-pass retrieved spoken segments a graph in which each node represents a segment and the edges represent the acoustic vector similarity between the segments' query hypotheses. Based on the concept that segments strongly connected to many segments with high/low scores on the graph should have higher/lower scores, the relevance scores for the segments propagate over the graph, and the segments

[1] Although the audio examples can be directly spoken by the user, they would be very mismatched to the target spoken archive, which certainly degrades the performance.

are re-ranked accordingly. In this way the spoken segments in the first-pass returned list are considered globally, rather than assuming a pseudo-relevant and -irrelevant set in the PRF approach. This approach is similar to the very successful PageRank [140,141] used to rank web pages; PageRank considers the hyperlink between every two pages and computes a converged importance score for each page. A similar approach has been found useful in video search, in which the similarity between each pair of videos is used to formulate the ranking problem over a graph [142,143].

Both example-based PRF and graph-based re-ranking based on acoustic similarity will be introduced in this chapter. In the previous chapters the proposed approaches were tested on a relatively limited task in which the query includes only a single in-vocabulary word, and the whole retrieval process was based on word lattices. Here the example-based approaches were formulated on a more complete task: the query can be shorter or longer, including one to several words, or can even contain OOV words, and the retrieval is considered on both word- and subword-based lattices.

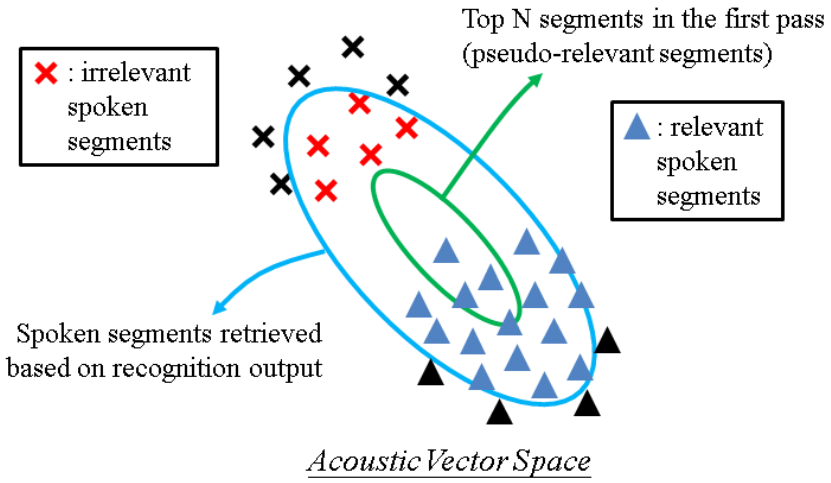


Figure 6.1: The demonstration for the concept of the example-based approaches.

6.2 Example-based Approaches

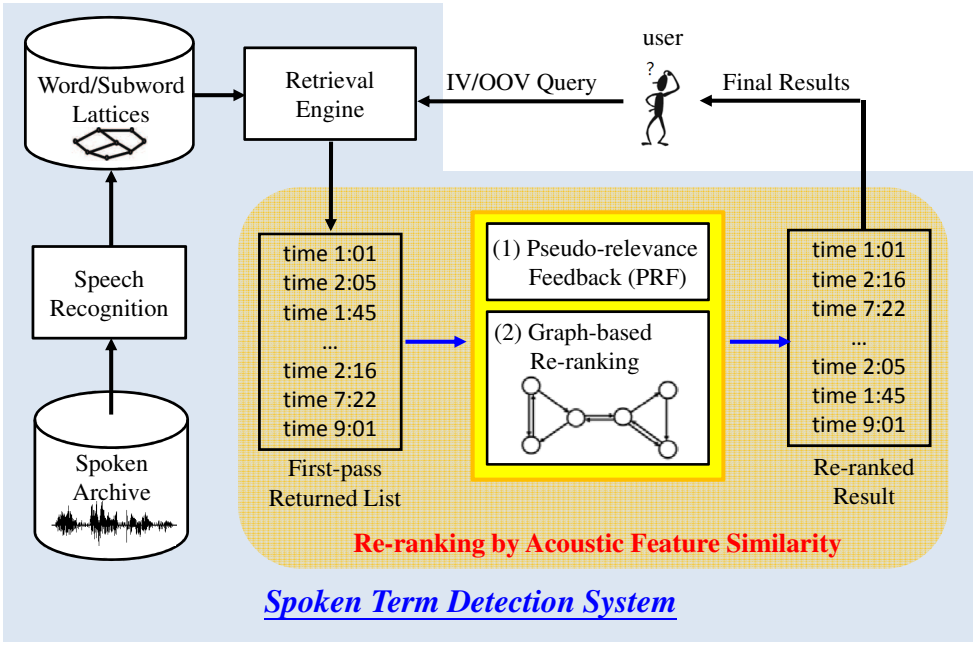


Figure 6.2: The complete framework for spoken content retrieval considering acoustic vector similarity.

The framework for the proposed approach for the task in question is shown in Fig. 6.2. The spoken segments are first transcribed into word or subword lattices by a speech recognizer. When the user enters a query, which can be shorter or longer including IV or OOV words, the retrieval engine searches over the lattices and produces the first-pass returned list as described in Section 6.2.1. The acoustic vector space similarity between every two retrieved segments is then computed as presented in Section 6.2.2. Based on these similarities, the list is re-ranked using either pseudo-relevance feedback (PRF) in Section 6.2.3 or graph-based re-ranking in Section 6.2.4.

6.2.1 Complete Formulation for the First-Pass Retrieval

As mentioned, given query Q, the system returns the spoken segments x_i with relevance scores higher than a threshold, and then ranks these segments according to the scores. However, the relevance score function $S(Q, x_i)$ in (4.1) does not support partial matching. For example, if the query Q is composed of three words $\{w_1, w_2, w_3\}$, only the spoken segments containing three concatenated arcs whose hypotheses are w_1, w_2 and w_3 respectively in their lattices can be retrieved. If one of the three words in the query is not correctly recognized in the lattices, it would not be returned, and this problem may decrease the recall rates of the system. To improve the performance, the system should also retrieve the lattices containing part of the query (for example, only $\{w_1, w_2\}$ or $\{w_2, w_3\}$ in the example), but give them smaller relevance scores.

Here a more complete relevance score function which allows partial matching is defined, which can be derived from either word or subword lattices, depending on which kinds of lattices are indexed. Relevance scores from word lattices are usually more accurate than those from subword lattices, but we must rely on the latter when the query Q consists of OOV words. Below we first show how to determine the new relevance scores using word lattices, and then show that the subword lattice-based scores can be obtained similarly.

We are given query Q consisting of one to several words, $Q = \{w_j, j = 1, 2, .., N_Q\}$, w_j being the j-th word and N_Q the number of words in Q. To compute the word-based relevance score $\hat{S}^{(w)}(Q, x_i)$ for a segment x_i from the word lattice, we calculate the expected count for each n-gram $\{w_k, ..., w_{k+n-1}\}$, $k = 1, ..., N_Q - n + 1$, in the query from the segment's lattice as in (6.1), and then aggregate the results for all such n-grams to

produce the score $S^{(w)}_{n-gram}(Q, x_i)$ for each order of n in (6.2).

$$E[w_k, ..., w_{k+n-1}|x_i] \tag{6.1}$$
$$= \frac{\sum_{u \in W(x_i)} P(x_i|u)P(u)N(u, \{w_k, ..., w_{k+n-1}\})}{\sum_{u \in W(x_i)} P(x_i|u)P(u)},$$

where $W(x_i)$ is the set of allowed paths in the lattice of x_i, u one of the allowed paths, $P(x_i|u)$ the likelihood for the observation sequence of x_i given the path u based on the acoustic model set, $P(u)$ the prior probability of u from the language model, and $N(u, \{w_k, ..., w_{k+n-1}\})$ the occurrence count of the n-gram $\{w_k, ..., w_{k+n-1}\}$ in u, and

$$S^{(w)}_{n-gram}(Q, x_i) = \sum_{k=1}^{N_Q-n+1} E[w_k, ..., w_{k+n-1}|x_i]. \tag{6.2}$$

The different proximity types, one for each n-gram order n allowed by the query length, are finally integrated in a weighted sum to yield the relevance score $\hat{S}^{(w)}(Q, x_i)$ for word lattices as

$$\hat{S}^{(w)}(Q, x_i) = \sum_{n=1}^{N_Q} a_n S^{(w)}_{n-gram}(Q, x_i), \tag{6.3}$$

where a_n is a weight parameter. Since $\hat{S}^{(w)}(Q, x_i)$ here is the aggregation of all the possible n-grams in the query, segments that only partially match the query can still be retrieved; this may increase the recall rate of the retrieval results but not necessary decrease the precision if a_n are properly set [37]. Note that $\hat{S}^{(w)}(Q, x_i)$ in (6.3) reduces to $S(Q, x_i)$ in (4.1) if the query Q consists of only a single word.

The subword-based relevance score $\hat{S}^{(s)}(Q, x_i)$ can be obtained in exactly the same way as that in (6.1) – (6.3), except that the query is represented as a sequence of subword units instead, $\{s_j, j = 1, 2, .., M_Q\}$, where s_j is the j-th subword unit and M_Q the number

of subword units in Q, and $E[s_k, ..., s_{k+n-1}|x_i]$ is computed on a subword lattice.

$$E[s_k, ..., s_{k+n-1}|x_i] \tag{6.4}$$
$$= \frac{\sum_{u \in W(x_i)} P(x_i|u)P(u)C(u, \{s_k, ..., s_{k+n-1}\})}{\sum_{u \in W(x_i)} P(x_i|u)P(u)},$$

$$S^{(s)}_{n-gram}(Q, x_i) = \sum_{k=1}^{M_Q - n + 1} E[s_k, ..., s_{k+n-1}|x_i], \tag{6.5}$$

$$\hat{S}^{(s)}(Q, x_i) = \sum_{n=1}^{M_Q} a'_n S^{(s)}_{n-gram}(x_i, Q). \tag{6.6}$$

Here (6.4), (6.5) and (6.6) are exactly the same as (6.1), (6.2) and (6.3) except that the word w_j is replaced by the subword unit s_j, $S^{(s)}_{n-gram}(Q, x_i)$ and $\hat{S}^{(s)}(Q, x_i)$ are subword-based n-gram score and subword-based relevance score respectively, and a'_n is the corresponding parameter.

6.2.2 Acoustic Vector Similarity

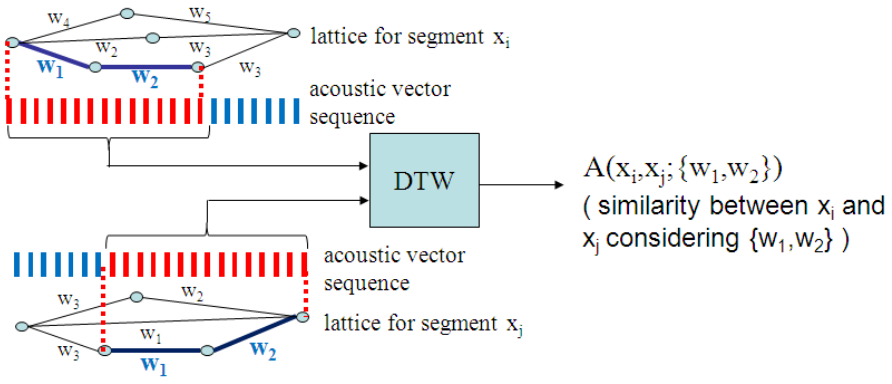


Figure 6.3: The computation of $A(x_i, x_j; \{w_1, w_2\})$, the acoustic vector similarity between x_i and x_j considering the 2-gram $\{w_1, w_2\}$.

Here the acoustic vector similarity between retrieved segments x_i and x_j is computed, which will be used in PRF and graph-based re-ranking in the next two subsections.

This similarity can be obtained again based on either word or subword units; here we show the word-based version first.

Given query Q consisting of a sequence of words $\{w_j, j = 1, 2, .., N_Q\}$, for each n-gram $\{w_k, ..., w_{k+n-1}\}$ in Q, the DTW distance [144] is first calculated between the acoustic vector sequences corresponding to the subpaths with word hypotheses $\{w_k, ..., w_{k+n-1}\}$ in the lattices of x_i and x_j[2]. An example is shown in Fig. 6.3. This yields $d(x_i, x_j; \{w_k, ..., w_{k+n-1}\})$, the DTW distance between x_i and x_j considering n-gram $\{w_k, ..., w_{k+n-1}\}$ in the query. The similarity between x_i and x_j considering $\{w_k, ..., w_{k+n-1}\}$ is then

$$\begin{aligned} & A(x_i, x_j; \{w_k, ..., w_{k+n-1}\}) \\ & = 1 - \frac{d(x_i, x_j; \{w_k, ..., w_{k+n-1}\}) - d_{min}}{d_{max} - d_{min}}, \end{aligned} \tag{6.7}$$

where d_{max} and d_{min} are the largest and smallest values of $d(x_i, x_j; \{w_k, ..., w_{k+n-1}\})$ for all pairs of segments in the first-pass returned list. Equation (6.7) simply normalizes the DTW distance and transforms it into a similarity score between 0 and 1. In the experiments in this chapter, MFCC vectors were used as the acoustic vectors. If the n-gram $\{w_k, ..., w_{k+n-1}\}$ does not exist in the lattice of either x_i or x_j, $A(x_i, x_j; \{w_k, ..., w_{k+n-1}\})$ is set to 0. We then aggregate the similarities considering all such n-grams to produce score $A^{(w)}_{n-gram}(x_i, x_j)$ for each order of n as

$$A^{(w)}_{n-gram}(x_i, x_j) = \sum_{k=1}^{N_Q - n + 1} A(x_i, x_j; \{w_k, ..., w_{k+n-1}\}). \tag{6.8}$$

The different proximity types are finally integrated as a weighted sum to yield the simi-

[2]If there are multiple subpaths whose word hypotheses are $\{w_k, ..., w_{k+n-1}\}$ in a lattice, only the one with the highest posterior probability is considered.

larity between x_i and x_j:

$$Sim^{(w)}(x_i, x_j) = \sum_{n=1}^{N} b_n A^{(w)}_{n-gram}(x_i, x_j), \tag{6.9}$$

where b_n is another weight parameter.

The computation of subword-based similarity $Sim^{(s)}(x_i, x_j)$ is exactly the same as that in (6.7) – (6.9), except that each word w_i is replaced by subword unit s_j.

$$\begin{aligned} &A(x_i, x_j; \{s_k, ..., s_{k+n-1}\}) \\ &= 1 - \frac{d(x_i, x_j; \{s_k, ..., s_{k+n-1}\}) - d'_{min}}{d'_{max} - d'_{min}}, \end{aligned} \tag{6.10}$$

$$A^{(s)}_{n-gram}(x_i, x_j) = \sum_{k=1}^{M_Q - n + 1} A(x_i, x_j; \{s_k, ..., s_{k+n-1}\}), \tag{6.11}$$

$$Sim^{(s)}(x_i, x_j) = \sum_{n=1}^{M_Q} b'_n A^{(s)}_{n-gram}(x_i, x_j). \tag{6.12}$$

Here (6.10), (6.11) and (6.12) are exactly the same as (6.7), (6.8) and (6.9), except the word w_j replaced by the subword unit s_j, $A^{(s)}_{n-gram}(x_i, x_j)$ and $Sim^{(s)}(x_i, x_j)$ are subword-based score and subword-based similarity respectively, and d'_{max}, d'_{min} and b'_n are the corresponding parameters.

Although we can obtain the relevance score and similarity based on different units and use them together – for example, it is possible to derive $\hat{S}^{(w)}(Q, x_i)$ in (6.3) from word lattices but compute $Sim^{(s)}(x_i, x_j)$ in (6.12) on subword lattices and use them together – for simplicity in the experiments below, we always use $\hat{S}^{(w)/(s)}(Q, x_i)$ and $Sim^{(w)/(s)}(x_i, x_j)$ obtained from the same type (word or subword) of lattices together. Below for simplicity in notation, we simply use $\hat{S}(Q, x_i)$ to denote relevance score and $Sim(x_i, x_j)$ to denote similarity, regardless of whether they are obtained from word or subword lattices.

6.2.3 Example-based Pseudo-relevance Feedback

In PRF, the top-ranked N_t segments with the highest relevance scores $\hat{S}(Q, x_i)$ are selected as pseudo-relevant set $\mathcal{Y}$; the bottom-ranked N_f segments with the lowest $\hat{S}(Q, x_i)$ are selected as pseudo-irrelevant set $\mathcal{Z}$. The similarity between each segment x_i in the first-pass result and the pseudo-relevant and -irrelevant sets is then defined as

$$PRF(x_i) = \frac{1}{N_t} \sum_{x \in \mathcal{Y}} Sim(x_i, x) - \frac{1}{N_f} \sum_{x \in \mathcal{Z}} Sim(x_i, x), \tag{6.13}$$

where N_t and N_f are the size of the pseudo-relevant and -irrelevant sets. The value of $PRF(x_i)$ is then linearly normalized between 0 and 1 as $PRF'(x_i)$. The relevance score $\hat{S}(Q, x_i)$ for each segment x_i is then updated as the new relevance score

$$S'_p(Q, x_i) = \hat{S}(Q, x_i)^{1-\delta_1} PRF'(x_i)^{\delta_1}, \tag{6.14}$$

where δ_1 is a weight parameter between 0 and 1. The segments retrieved are then re-ranked according to $S'_p(Q, x_i)$, and then displayed to the user.

6.2.4 Graph-based Re-ranking

An alternative to PRF for calculating acoustic vector similarity is graph-based re-ranking, which involves first constructing a graph for the retrieved segments and then applying a random walk for relevance score propagation.

Here a directed graph such as that in Fig. 6.4 is constructed from the first-pass returned list, in which each node represents a segment. The weight for the edge from x_i to x_j ($x_i \rightarrow x_j$) is defined as $Sim(x_i, x_j)$. For the graph-based approach, usually the edges with small weights would be pruned to obtain better results.

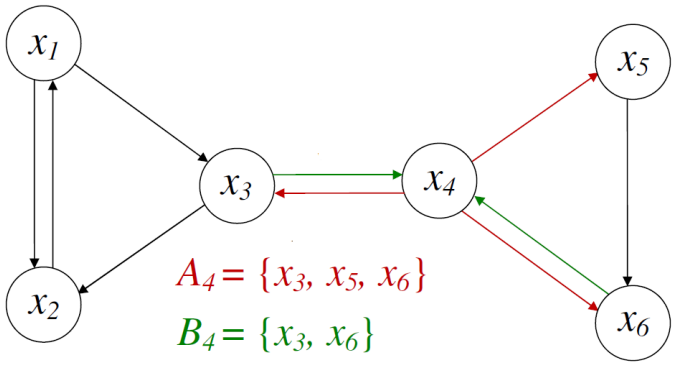


Figure 6.4: A simplified example of a graph, the nodes of which correspond to segments. The edge weights are acoustic similarities between the nodes. A_i and B_i are the node sets connected respectively by the outgoing and incoming edges of x_i.

A new set of graph-based relevance scores $S_g(Q, x_i)$ for all x_i in the first-pass returned list is then obtained via score propagation on the graph, which satisfies

$$S_g(Q, x_i) = (1 - \alpha)\hat{S}(Q, x_i) + \alpha \sum_{x_j \in B_i} S_g(Q, x_j) Sim'(x_j, x_i), \tag{6.15}$$

where $\hat{S}(Q, x_i)$ is the relevance score in (6.3), α is an interpolation weight between 0 and 1, B_i is the set of all segments connected to x_i as in Fig. 6.4, and x_j is a node connected to x_i ($x_j \rightarrow x_i$). $Sim'(x_j, x_i)$ is the normalized edge weight $Sim(x_j, x_i)$ over the edges that start from node x_j on the graph:

$$Sim'(x_j, x_i) = \frac{Sim(x_j, x_i)}{\sum_{x_k \in A_j} Sim(x_j, x_k)}, \tag{6.16}$$

where A_j is the set of edges that start from x_j as in Fig. 6.4. In (6.15) the graph-based score $S_g(Q, x_i)$ of a segment x_i depends on two factors interpolated by α: the relevance score $\hat{S}(Q, x_i)$ (the first term on the right side of (6.15)) and the score propagation over the graph based on the normalized edge weights $Sim'(x_j, x_i)$ (the second term on the right side). Based on (6.15), $S_g(Q, x_i)$ would be large if $\hat{S}(Q, x_i)$ is large, or x_i strongly connected to many segments x_j with large $S_g(Q, x_j)$ on the graph. The normalization

in (6.16) formulates (6.15) as a random walk problem on the graph; random walk theory guarantees that a set of unique solutions of $S_g(Q, x_i)$ can be found since the random walk here is actually formulated on an irreducible and aperiodic graph [140].

$S_g(Q, x_i)$ can be found by power method efficiently. Each node x_i is first assigned an initial value $S_g^0(Q, x_i)$ (the initial values would not influence the final solutions). At the t-th iteration, $S_g^t(Q, x_i)$ is obtained as follows.

$$S_g^t(Q, x_i) = (1 - \alpha)\hat{S}(Q, x_i) + \alpha \sum_{x_j \in B_i} S_g^{t-1}(Q, x_j) Sim'(x_j, x_i). \tag{6.17}$$

When the process finally converges ($S_g^t(Q, x_i) \approx S_g^{t-1}(Q, x_i)$), $S_g^t(Q, x_i)$ can be taken as $S_g(Q, x_i)$.

$S_g(Q, x_i)$ is finally integrated with $\hat{S}(Q, x_i)$ for re-ranking as

$$S'_g(Q, x_i) = \hat{S}(Q, x_i)^{1-\delta_2} S_g(Q, x_i)^{\delta_2}, \tag{6.18}$$

where δ_2 is a parameter between 0 and 1. The final retrieval results ranked according to $S'_g(Q, x_i)$ in (6.18) are then displayed to the user.

6.3 Experiments for Lecture Courses

6.3.1 IV queries

The results for a set of IV queries on the lecture courses are reported below.

Experimental Setup

In the following experiments, δ_1 in (6.14) and δ_2 in (6.18) were both set to 0.9, a_n and a'_n in (6.3) and (6.6) were both set to 10^{5n} to favor longer n-grams, and b_n and b'_n in (6.9) and

(6.12) were equal to a_n and a'_n. Mean average precision (MAP) was used as the retrieval performance measure.

The testing spoken archive here is the recorded lectures used in the last two chapters. In order to evaluate the retrieval performance with respect to acoustic models of different matched conditions, we used three sets of acoustic models. For all three sets of acoustic models, we trained a set of state-tied triphones spanned from 37 Mandarin monophones and 35 English monophones based on the recently-developed state mapping and recovery techniques [145], so they were different from the acoustic models used in the previous chapters. The three acoustic models are

- Speaker-independent models (SI) trained on a Mandarin corpus of 24.6 hours of read speech, produced by 100 male and 100 female speakers, plus the Sinica L2 Taiwanese English corpus with 59.7 hours of English read speech, produced by 229 male and 256 female Taiwanese speakers.

- Speaker-adaptive models (SA) adapted by MLLR with 256 classes cascaded with the maximum a posterior estimation from the above SI model based on 500 utterances taken from the training set of the lecture corpus.

- Speaker-dependent models (SD) trained on the 12-hour data which came from the course of the same instructor but different from the testing archive here.

The testing query set included 275 Chinese queries composed of 1 to 3 words, or 2 to 7 Chinese characters. In the experiments here, the language model was trained with the manual transcriptions of the 12-hour training set of the lecture corpus. A close-to-oracle lexicon was used which included 11K Chinese words plus 2K English words covering all

words in the testing archive. Each utterance was transcribed into a bilingual word lattice. Then we transformed each Chinese word arc into a sequence of concatenated Chinese character or Mandarin syllable arcs to respectively form character or syllable lattices. Therefore, for each utterance there were three lattices: word-, character-, and syllable-based. The recognition accuracies (character accuracies for Mandarin Chinese and word accuracies for English) were 49.7%, 80.8%, and 88.0% respectively for the SI, SA, and SD models.

Example-based Pseudo-relevance Feedback

Table 6.1 shows the MAP performance yielded by PRF with different numbers of pseudo-relevant segments (different N_t in (6.13)) and 40 pseudo-irrelevant segments ($N_f = 40$ in (6.13)). The first-pass retrieval results are considered as the baselines. The three columns SI, SA and SD correspond to three acoustic models with different qualities. First of all, we found that PRF outperformed the baselines except when $N_t = 1$. We also observed that as the number of pseudo-relevant segments was raised the MAP first increased and then decreased. This is reasonable because a larger N_t implies that more segments are considered when computing the similarities, and thus that disturbances caused by incorrect assumptions about segment relevance (irrelevant segments assumed to be relevant) are diluted. However, when N_t was too large, since a query usually only has few relevant segments in a spoken archive, more irrelevant segments were inevitably included, naturally degrading the MAP.

Table 6.2 shows the performance with the number of pseudo-relevant segments fixed at 9 ($N_t = 9$ in (6.13)) but with different numbers of pseudo-irrelevant segments (vari-

Table 6.1: The MAP performance of PRF with different numbers of pseudo-relevant segments and 40 pseudo-irrelevant segments. The first-pass retrieval results are considered as the baselines. SI, SA, and SD correspond to the three acoustic models.

Acoustic model		SI	SA	SD
Baseline		0.5596	0.7956	0.8424
Number of pseudo-relevant segments	1	0.5972	0.7946	0.8392
	3	0.6146	0.8126	0.8529
	5	0.6199	0.8204	0.8583
	7	0.6223	0.8216	0.8605
	9	0.6235	0.8220	0.8601
	11	0.6208	0.8205	0.8611
	13	0.6219	0.8192	0.8606
	15	0.6185	0.8172	0.8591
	17	0.6157	0.8157	0.8574
	19	0.6136	0.8149	0.8557

ous N_f). In contrast to Table 6.1, we observed that as the number of pseudo-irrelevant segments was raised the MAP first increased and then saturated without too much degradation. This may be because the irrelevant segments usually form the majority of the retrieved segments, so most pseudo-irrelevant segments are truly irrelevant even for very large values of N_f.

Table 6.2: MAP performance of PRF with 9 pseudo-relevant segments but different numbers of pseudo-irrelevant segments.

Acoustic model		SI	SA	SD
Baseline		0.5596	0.7956	0.8424
Number of pseudo-irrelevant segments	5	0.6083	0.8174	0.8557
	10	0.6150	0.8194	0.8569
	15	0.6197	0.8196	0.8580
	20	0.6222	0.8215	0.8579
	25	0.6228	0.8221	0.8602
	30	0.6234	0.8225	0.8599
	35	0.6235	0.8222	0.8603
	40	0.6235	0.8220	0.8601
	45	0.6235	0.8218	0.8600

Graph-based Re-ranking

In the experiments for the lecture courses, the graphs were all constructed with the following strategy. Each segment (or node) x_i was connected by the K' segments x_j with highest $Sim(x_i, x_j)$ ($x_i \leftarrow x_j$). Thus each node in the graph had a fixed number of incoming edges but a variable number of outgoing edges. Table 6.3 shows the results of graph-based re-ranking yielded by different numbers of incoming edges K' with three sets of different acoustic models, one per column. We found that graph-based re-ranking outperformed the baselines except when $K' = 1$. The PRF results are also reported; the numbers of pseudo-relevant and -irrelevant segments here were tuned to maximize the

Table 6.3: MAP of graph-based re-ranking for different numbers of **incoming edges** K' using different acoustic models. The best results in each column are in bold.

Acoustic model		SI	SA	SD
Baseline		0.5596	0.7956	0.8424
PRF		0.6261	0.8239	0.8621
Graph	$K' = 1$	0.5873	0.7884	0.8315
	$K' = 2$	0.6463	0.8127	0.8566
	$K' = 3$	0.6679	0.8239	0.8666
	$K' = 4$	0.6753	0.8281	0.8690
	$K' = 5$	**0.6783**	0.8328	0.8711
	$K' = 10$	0.6699	**0.8337**	**0.8717**
	$K' = 15$	0.6612	0.8301	0.8700
	$K' = 20$	0.6550	0.8271	0.8678

MAP values on the testing query set, resulting in unrealistically high performance for PRF. We found that graph-based re-ranking was so powerful that even though the parameters for PRF were tuned in this fashion, graph-based re-ranked still outperformed PRF if K' was reasonably large.

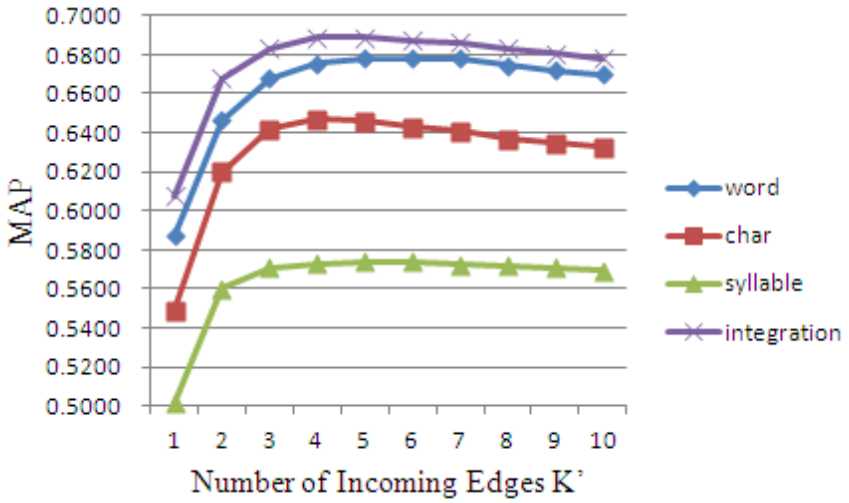
Experimental Results based on Subword Lattices

Table 6.4 shows the results with word and subword units. Parts (a), (b), and (c) are respectively for word-, character- or syllable-based retrieval, and columns SI, SA, and SD correspond to the different acoustic models. For each case we report the results of the first

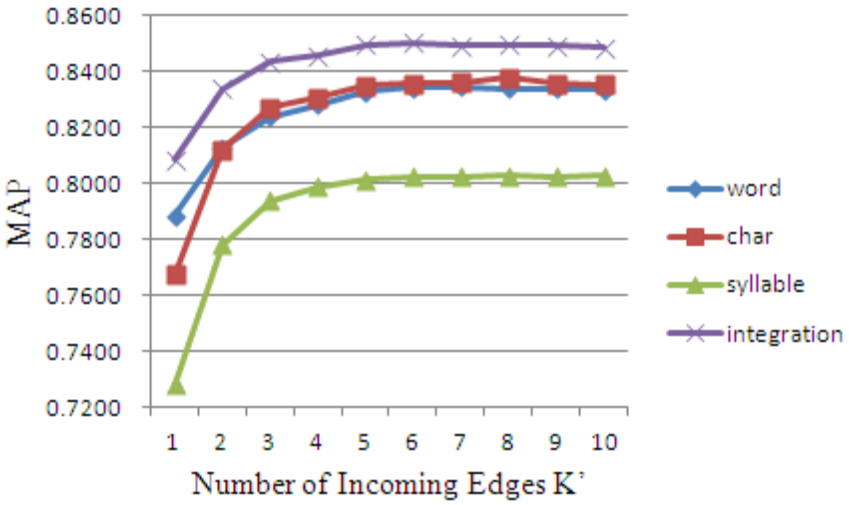
Table 6.4: MAP results of first pass (baseline), PRF, and graph-based re-ranking under different acoustic models with word-, character-, or syllable-based retrieval.

	Approach	SI	SA	SD
(a) *Word*	Baseline	0.5596	0.7956	0.8424
	PRF	0.6261	0.8239	0.8621
	Graph	0.6783	0.8328	0.8711
(b) *Character*	Baseline	0.4733	0.7216	0.7507
	PRF	0.5761	0.8209	0.8595
	Graph	0.6462	0.8349	0.8666
(c) *Syllable*	Baseline	0.4329	0.6737	0.6941
	PRF	0.5281	0.7797	0.8182
	Graph	0.5739	0.8014	0.8308

pass (baseline), PRF, and graph-based re-ranking. Again, the numbers of pseudo-relevant and -irrelevant segments for PRF were tuned on the testing queries. It is clear that PRF always yields improvements over the baseline, and that graph-based re-ranking always yields still further improvements regardless of the acoustic model or unit type. Note also that even though the word-based first-pass results were much better than the subword-based results, PRF and graph-based re-ranking yielded larger improvements for subword lattices. Because both PRF and graph-based re-ranking only re-rank the first-pass retrieval results, spoken segments that were not retrieved in the first pass are never retrieved. Therefore, since subwords have higher recall than words, the proposed approach yielded greater improvements for subwords.

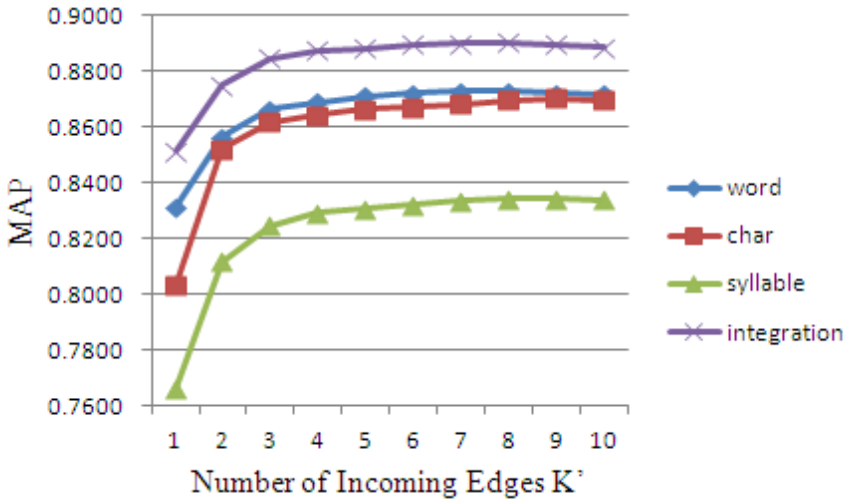


(a) SI model



(b) SA model

Figure 6.5: MAP performance of the graph-based re-ranking based on word, character, syllable, and the integration of the three different units. The horizontal scales in the figures are the numbers of incoming edges. (a), (b) and (c) are respectively for SI, SA and SD models.



(c) SD model

Figure 6.5: MAP performance of the graph-based re-ranking based on word, character, syllable, and the integration of the three different units. The horizontal scales in the figures are the numbers of incoming edges. (a), (b) and (c) are respectively for SI, SA and SD models.

Fig. 6.5 is the MAP performance of graph-based re-ranking based on word, character, syllable, and their integration. The horizontal scales in the figures are the numbers of incoming edges K'. Fig. 6.5 (a), (b), and (c) are respectively for SI, SA, and SD models. Integration was achieved as a weighted sum of the relevance scores of the results obtained based on word, character, and syllable, the weights of which were 1.0, 0.2 and 0.04 respectively. Since different units contain complementary information, the integration of the results of different units after graph-based re-ranking always outperformed each individual lattices regardless of the acoustic models or the number of the incoming edges K'.

6.3.2 OOV queries

In this section, the example-based PRF and graph-based re-ranking introduced were tested on a set of OOV queries. To handle the problem of OOV, the entered queries were transformed into a sequence of subword units via the grapheme-to-phoneme technique, and the relevance scores $\hat{S}^{(s)}(Q, x_i)$ in (6.6) and similarity $Sim^{(s)}(x_i, x_j)$ in (6.12) were computed based on the lattices with subword hypotheses.

Experimental Setup

The testing spoken archive for the OOV queries is also the recorded lectures. For the OOV query set we used 110 English queries, each consisting of a single word. We trained a 6-gram joint-sequence model from the CMU dictionary with 130K words as the grapheme-to-phoneme converter to predict pronunciations for OOV queries [146][3]. The canonical pronunciation for each OOV query was also used. Using the canonical pronunciation as

[3]The terms in OOV queries were excluded from the CMU dictionary during training.

the reference, the accuracies of the estimated pronunciation on syllable and phone levels were 85.8% and 93.8% respectively. The pronunciation was estimated perfectly (same as the reference) for 81 of the 110 OOV queries (73.6%).

A word/subword hybrid system was used to transcribe each spoken segment. This kind of hybrid system have been used to handle the OOV problem [147]. In this system, we used a lexicon composed of 11K Chinese words, 5K English words from the standard Aurora-4 lexicon, and 10K English syllables for recognition (the English queries were not included in the 5K English words). 20,000 English documents from the 20Newsgroups corpus[4] were used to train the English language model, which was a mix of word and syllable n-grams. Those words in the English documents that were not included in the 5K English words were transformed into their corresponding syllable sequences, which we used to train an English word and syllable mixed trigram language model. A Chinese word-based trigram language model was trained on the manual transcriptions of the 12-hour training set of the lecture corpus. These two language models were then interpolated to produce the lattices composed of a mixture of Chinese words, English words, and English syllable arcs. We further substituted the Chinese and English word arcs in the lattices with their corresponding syllables to obtain a set of syllable-based lattices. Thus for each spoken segment we generated two lattices: one composed of Chinese and English words and English syllables, and the other composed solely of Chinese and English syllables. Since the English recognition accuracies for SI and SA were not good enough to obtain reasonable results, we used only SD models in Section 6.3.1 for the OOV query experiments. Since word accuracy for the hybrid recognition output is undefined, we evaluated

[4]http://people.csail.mit.edu/jrennie/20Newsgroups/

the English syllable accuracy instead, which was 43.6% for the SD models.

Experimental results

Here we applied example-based PRF and the graph-based re-ranking on the OOV query set to determine if these approaches do well with OOV queries too. For the graph construction, here each segment is connected by the K' segments with highest acoustic similarities, so each node in the graph has a fixed number of incoming edges but a variable number of outgoing edges. Table 6.5 shows the results for the OOV query experiments on the SD-generated lattices. Section (a) is for the canonical pronunciation of each OOV query, while section (b) shows the results based on the pronunciation estimated using the grapheme-to-phoneme approach. Columns (1) and (2) are respectively for the two sets of lattices described in Section 6.3.2. Column (1) (Hybrid) refers to the lattices composed of Chinese and English word arcs as well as English syllable arcs, while column (2) (Syllable) refers to the lattices composed only of Chinese and English syllable arcs. The numbers of pseudo-relevant and -irrelevant segments for PRF were determined by 4-fold cross validation here. That is, the testing queries were first separated into 4 folds. Each fold was selected once as the development query set for parameter tuning with the other folds set aside as the testing query set.

We found that the grapheme-to-phoneme-based pronunciations yielded lower performance compared with the canonical pronunciation (section (b) vs (a)). Also, the lattices purely composed of syllables (column (2)) outperformed the hybrid lattices (column (1)). Because some of the English queries were wrongly recognized as words with similar pronunciations in the lexicon, transforming those words into corresponding syllable

sequences increased the recall rates and thus improved the results.

We found that remarkable improvements were achieved by both PRF and graph-based re-ranking in all cases. However, we also observed that the graph-based re-ranking did not outperform PRF on the OOV query set. This may be because due to the poor recognition results for the OOV terms, the relevance scores $\hat{S}^{(s)}(Q, x_i)$ in (6.6) might be unreliable. Since the graph-based re-ranking in (6.15) directly applied the lattice-derived relevance scores, it may be relatively sensitive to the noisy relevance scores from the first pass. On the other hand, PRF considered the ranking of the first pass, which may be more robust since the disturbance of the scores did not always imply the change of the ranking positions. However, more evidences from different corpora are necessary to further verify this conclusion.

6.4 Experiments for Broadcast News

6.4.1 Experimental Setup

Then example-based PRF and the graph-based approach were tested on the Mandarin broadcast news used in the previous chapters. In order to evaluate the retrieval performance of the proposed approaches with respect to different recognition conditions, different acoustic and language models were used to transcribe the spoken archive. As listed below, two different recognition conditions were used for the spoken archive:

- Archive (A): The recognition condition used in Section 4.6.1.
- Archive (B): Perceptual Linear Predictive (PLP) feature and phone posterior probabilities estimated by a Multilayer Perceptron (MLP) trained from 10 hours of broad-

cast news were cascaded as the acoustic vectors. A tri-gram language model trained on 98.5M words of news from several different sources, and a set of acoustic models with 48 Gaussian mixtures per state and 3 states per model trained on the 24.5 hours of broadcast news were used. The character accuracy was 62.13%.

The beam width for recognition was both 100 for Archives (A) and (B).

6.4.2 Experimental Results

Table 6.6 shows the results of example-based PRF and the graph-based approach for both Archives (A) and (B). The superscript labels * and † respectively indicate significantly better than the baseline and example-based PRF. $\hat{S}^{(w)}(Q, x_i)$ in (6.3) was used as the relevance score. For the graph construction, the spoken segments x_i and x_j are connected to each other if x_i is among the K'-nearest neighbors of x_j based on $Sim^{(w)}(x_i, x_j)$ in (6.9), and x_j is among the K'-nearest neighbors of x_i. The sizes of pseudo-relevant and -irrelevant spoken segments (N_t and N_f in (6.13)), the value of K' and α in (6.15) were determined by 4-fold cross validation. That is, the testing queries were first separated into 4 folds. Each fold was selected once as the development query set for parameter tuning with the other folds set aside as the testing query set. Clearly, all the example-based approach obtained significant improvements over the baselines, and the graph-based approach outperformed PRF, which was consistent with the results observed on the lecture courses.

Finally, in Table 6.7, the best result for acoustic model re-estimation method in short-term context ($N = 15$ in Table 4.6), and the best result for SVM with PRF (enhanced PRF with $K = 10$ and $N' = 30$ in Figure 5.6) are compared with the example-based

methods in Table 6.7 on Archive (A). It looks like example-based approaches in rows (4) and (5) were better than the SVM-based PRF in row (3). This may be because SVM-based PRF needed state boundaries which may be inaccurate with poor recognition models, and all the acoustic vectors in the same states were simply averaged. On the other hand, example-based approaches were based on DTW considering all the information embedded in the acoustic vector sequences. SVM is able to give different dimensions of the acoustic vectors in different states different importance, which was not considered in those example-based methods, but this benefit was not reflected in the performance. The model re-estimation method was the worst among these approaches even though it was based on user relevance feedback with correct relevance information, and acoustic model re-estimation did not obtain improvements in the PRF scenario. Since MAP values are often dominated by the top several items selected, the improvements for user relevance feedback in MAP scores were actually limited by the frozen, so it is hard to compare the results based on user relevance feedback and PRF. The inherent problem of the acoustic model re-estimation approach has been discussed. Since there were too many parameters in the acoustic models, but only limited training data was available from relevance feedback, the model re-estimation process was actually very risky, and some regularization techniques may be necessary here. It is clear that the graph-based approach was the best among these approaches. However, it is too arbitrary to conclude that the example-based approach is the best method, since all of the approaches have some rooms for improvement. This will be further discussed in the last chapter.

6.5 Summary

In this chapter, the example-based approaches including example-based PRF and the graph-based approach both taking into account acoustic vector similarity were developed and tested, and these approaches were applied to spoken term detection with both IV and OOV queries. We found that these approaches can not only yield improved performance for word-based lattices but also for subword-based lattices.

Table 6.5: MAP for OOV queries on SD-generated lattices for different pronunciations, lattice types, and number of incoming edges K'.

		(a) Canonical		(b) g2p	
		(1) Hybrid	(2) Syllable	(1) Hybrid	(2) Syllable
Baseline		0.3611	0.3806	0.3092	0.3288
PRF		0.4699	0.4967	0.4127	0.4362
Graph	$K' = 1$	0.4246	0.4423	0.3621	0.3888
	$K' = 2$	0.4504	0.4659	0.3874	0.4177
	$K' = 3$	0.4613	0.4697	0.4020	0.4232
	$K' = 4$	0.4666	0.4757	0.4087	0.4287
	$K' = 5$	0.4654	0.4760	0.4087	0.4293
	$K' = 10$	0.4644	0.4775	0.4114	0.4320
	$K' = 15$	0.4684	0.4794	0.4175	0.4340
	$K' = 20$	0.4742	0.4823	0.4215	0.4349
	$K' = 100$	0.4766	0.4840	0.4213	0.4328

Approach	Archive (A)	Archive (B)
Baseline	0.6302	0.6651
Example-based PRF	0.6577*	0.6937*
Graph-based Approach	0.6685*†	0.6976*

Table 6.6: Experimental results for Archives (A) and (B) for example-based PRF and the graph-based approach. The baselines are the results without feedback. The superscript labels * and † respectively indicate significantly better than the baseline and example-based PRF.

Approach	MAP
(1) Baseline	0.6302
(2) Acoustic Model Re-estimation in Short-term Context (Section 4.3)	0.6482
(3) SVM with Enhanced PRF (Section 5.4)	0.6572
(4) Example-based PRF (Section 6.2.3)	0.6577
(5) Graph-based Approach (Section 6.2.4)	0.6685

Table 6.7: The comparison of different methods on Archive (A) in terms of MAP. The baseline is the result without relevance feedback.

Chapter 7 Semantic Retrieval for Spoken Content with Acoustic Similarity Graph

7.1 Introduction

Most works in spoken content retrieval nowadays continue to focus on spoken term detection, for which the goal is simply returning spoken segments that contain the query terms. This is insufficient because users naturally prefer that the technologies can return all the objects that the users really want, regardless of whether the query terms are contained or not. Therefore, there have been some recent works on semantic retrieval for spoken content [148–153].

In text-based information retrieval, even if the texts to be retrieved include all precise words, it is still difficult to retrieve all documents relevant to the query, because many queries are too short to completely represent the user's intent, but the techniques for semantic retrieval such as latent semantic analysis [154–156] and query expansion [101, 106,157] have been widely studied in text-based information retrieval. Taking ASR transcriptions as text, these techniques developed for text-based information retrieval can be directly applied on spoken content retrieval. However, since these techniques were developed for text without errors, the inevitable recognition errors in ASR transcriptions may degrade the performance. It has been found that enhancing the estimation of the term frequencies in a spoken document based on the context information of the query hypotheses improves the performance of both language modelling retrieval approach and query expansion [148].

In the previous chapters, it has been found that acoustic vector similarity between spoken segments is very helpful for the spoken term detection task with the graph-based re-ranking approach. In this chapter, to have more robust semantic retrieval for spoken documents, the expected term frequencies derived from the lattices are enhanced by acoustic similarity with the graph-based approach. The enhanced expected term frequencies can not only improve the performance of the language modelling retrieval approach, but also boosts the performances of the document expansion techniques based on latent semantic analysis, and query expansion methods considering both words and latent topic information. Good improvements were observed in the preliminary experiments.

7.2 Language Modelling for Spoken Content Retrieval

Language modelling approach has been known to be very effective for information retrieval not only for text, but for spoken content as well [158]. The basic idea for language modelling approach is that both queries and documents are respectively represented as language models θ_Q and θ_d, and the relevance score function $S_L(Q, d)$ used for ranking the documents d for query Q is the inverse of the KL divergence between θ_Q and θ_d:

$$S_L(Q, d) = -KL(\theta_Q|\theta_d). \tag{7.1}$$

That is, the documents whose language models are similar to the model of the query are more likely to be relevant. In this way, the problem of retrieval is reduced to the estimation of the language models for the queries and documents. To simplify the presentation, here we assume word unigram language models for documents and queries only, although the proposed approach is not limited to this case.

Usually the language model θ_Q is estimated based on the words in the query in (7.2).

$$P(w|\theta_Q) = \frac{N(w,Q)}{|Q|}, \tag{7.2}$$

where $P(w|\theta_Q)$ is the probability of generating the word w from the model θ_Q, $N(w,Q)$ the occurrence counts of the word w in query Q, and $|Q|$ the total number of words in the query. For document language model θ_d, when the spoken documents are transcribed into 1-best transcriptions, the estimation for θ_d is just exactly the same as text-based information retrieval. However, as there are inevitable relatively high recognition errors in the 1-best transcriptions, θ_d thus estimated may be very different from the true word distribution of the spoken document. In Section 7.2.1, the latticed-derived document language model is represented, and the document model is further enhanced based on the acoustic similarity with the graph-based approach in Section 7.2.2. Both the document model θ_d and query model θ_Q will be semantically expanded in the following two sections.

7.2.1 Lattice-derived Document Model

Each spoken document d in the collection is first divided into spoken segments $\{x_1, \cdots, x_i, \cdots, x_I\}$, where I is the number of spoken segments in d, and then each spoken segment x_i is transcribed into a lattice. We first compute the expected counts of each word w on the lattice of each segment x_i:

$$E[w|x_i] = \sum_{u \in W(x_i)} N(w,u) P(u|x_i), \tag{7.3}$$

where u is a word sequence in the lattice, $W(x_i)$ is the set of all possible word sequences in the lattice for x_i, $N(w,u)$ is the occurrence counts of the word w in u, and $P(u|x_i)$ is the posterior probability of the word sequence u derived from the acoustic and language

models.

$$P(u|x_i) = \frac{P(u|x_i)P(u)}{\sum_{u \in W(x_i)} P(u|x_i)P(u)}, \tag{7.4}$$

where $P(x_i|u)$ the likelihood for the observation sequence of x_i given the path u based on the acoustic model set, and $P(u)$ the prior probability of u from the language model.

The language model $\theta^l_{x_i}$ for each spoken segment x_i is estimated in (7.5) [1].

$$P(w|\theta^l_{x_i}) = \frac{E[w|x_i]}{L_{x_i}}, \tag{7.5}$$

where L_{x_i} is the expected length for segment x_i,

$$L_{x_i} = \sum_{u \in W(x_i)} |u| P(u|x_i), \tag{7.6}$$

in which $|u|$ is the number of word arcs in u. All the language models $\theta^l_{x_i}$ for the spoken segments x_i in the document d are interpolated based on their expected lengths L_{x_i} to form a document model θ^l_d in (7.7).

$$P(w|\theta^l_d) = \frac{\sum_{i=1}^{I} L_{x_i} P(w|\theta^l_{x_i})}{L_d}, \tag{7.7}$$

where L_d is the expected document length which is the sum of the length of all the segments in the documents, or $L_d = \sum_{i=1}^{I} L_{x_i}$.

Then θ^l_d is interpolated with a background language model θ_b trained from all the spoken documents in the collection $\mathcal{C}$ to form a smoothed model $\bar{\theta}^l_d$ in (7.8).

$$P(w|\bar{\theta}^l_d) = a_d P(w|\theta^l_d) + (1 - a_d) P(w|\theta_b), \tag{7.8}$$

where

$$P(w|\theta_b) = \frac{\sum_{d \in \mathcal{C}} L_d P(w|\theta^l_d)}{\sum_{d \in \mathcal{C}} L_d}, \tag{7.9}$$

[1] The superscript ^{l} indicates that the language models are derived from the lattices.

which is the probability of observing the word w in the whole collection $\mathcal{C}$. a_d in (7.8) is a document dependent interpolation weight, which is equal to $\frac{L_d}{L_d+a}$ (a is a parameter to be set). In this way, the background model would have more influence on the shorter documents [159]. Finally, the smoothed model $\bar{\theta}_d^l$ is served as θ_d in (7.1) for ranking. Very similar formulation has been proposed for document model estimation based on the lattices [158].

It is also possible to apply the BM25 [160] model for semantic retrieval of spoken content. However, this model applies the inverse document frequency of each term w in (7.10) when computing the relevance scores.

$$IDF(w) = log(\frac{|\mathcal{C}|}{N_w}), \tag{7.10}$$

where $|\mathcal{C}|$ is the total number of documents in the whole collection, and N_w is the number of documents containing word w. Computing N_w on text is trivial, but whether the word w exists in a spoken document can not be directly observed. Although it is possible to solve this problem heuristically, it may result in extra parameters to be tuned. Therefore, BM25 is not considered in this thesis.

7.2.2 Acoustic Similarity Enhanced Document Model

Although the document models derived from the lattices may be better than the ones based on the 1-best transcriptions, they still unavoidably suffer from the recognition errors. Considering an arc with word hypothesis w in the lattice of a spoken segment, if its corresponding acoustic vector sequence is similar to those arcs which are also recognized as word hypothesis w in other spoken segments, this word hypothesis will be more trust-worthy; otherwise it may be suspected. Based on this concept, the expected counts

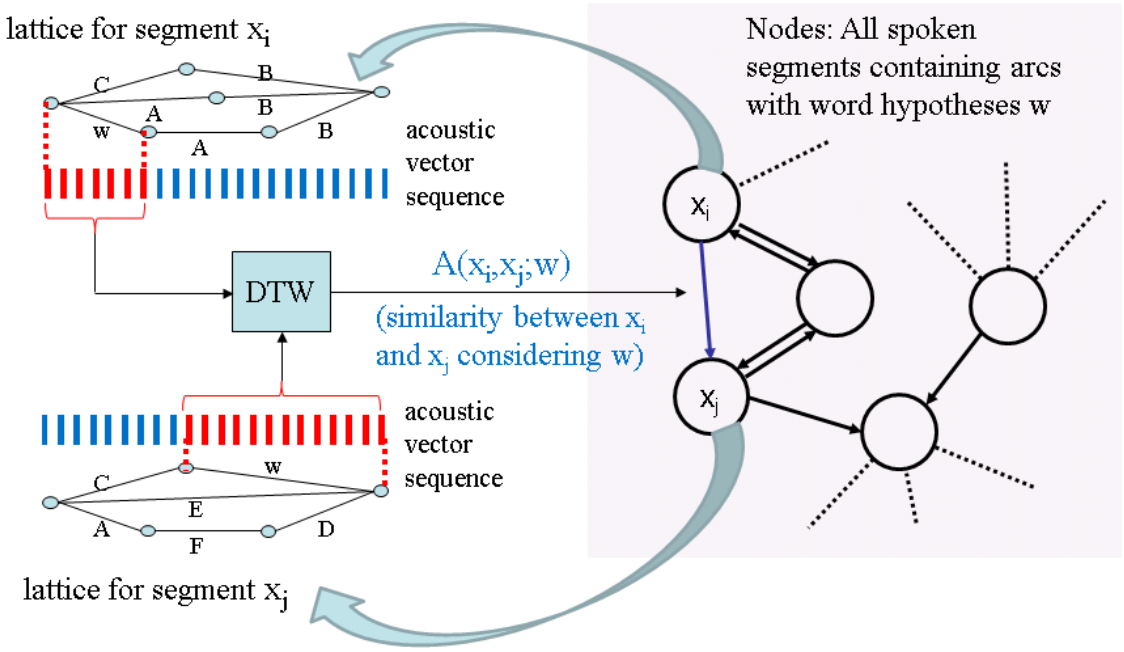


Figure 7.1: The graph constructed for computing the enhanced expected counts for word w based on acoustic similarity. Nodes in the graph are all spoken segments containing word arc w in the lattices, and the edge weights represent the acoustic similarities between the nodes considering word w.

$E[w|x]$ derived from the lattices can be enhanced by the graph-based approach.

The graph-based approach used here is nearly the same as the one introduced in Chapter 6, expect that it is conducted on all the words w in the lexicon. In the graph-based approach, for all the words w in the lexicon, we first construct a graph based on all the segments containing arcs with word hypotheses w in the lattices as Fig 7.1, in which each node represents a spoken segment. The dynamic time warping (DTW) distance is calculated between the acoustic vector sequences corresponding to the word hypotheses w in all segment pairs x_i and x_j on the graph [2]. This yields $d(x_i, x_j; w)$, the DTW distance

[2]If there are multiple arcs whose word hypotheses are w in a lattice, only the one with the highest posterior probability is considered as well. Although this assumption neglects the case that a word may

between x_i and x_j considering the term w. The similarity between x_i and x_j considering w is then

$$A(x_i, x_j; w) = 1 - \frac{d(x_i, x_j; w) - d_{min}}{d_{max} - d_{min}}, \tag{7.11}$$

where d_{max} and d_{min} are the largest and smallest values of $d(x_i, x_j; w)$ for all pairs of segments on the graph. Then $A(x_i, x_j; w)$ is taken as the weight of edge from x_i to x_j.

The acoustic similarity enhanced expected counts $E_a[w|x_i]$ for word w in segment x_i are then obtained via score propagation on the graph, which makes $E_a[w|x_i]$ satisfy [3]

$$E_a[w|x_i] = (1-\alpha)E[w|x_i] + \alpha \sum_{x_j \in B_i} E_a[w|x_j]\hat{A}(x_j, x_i; w), \tag{7.12}$$

where $E[w|x_i]$ is the lattice-derived expected counts in (7.3), α is an interpolation weight between 0 and 1, B_i is the set of all segments connected to x_i, and x_j is a node connected to x_i. $\hat{A}(x_j, x_i; w)$ is the normalized edge weight $S(x_j, x_i; w)$ over the edges that start from node x_j on the graph:

$$\hat{A}(x_j, x_i; w) = \frac{A(x_j, x_i; w)}{\sum_{x_k \in A_j} A(x_j, x_k; w)}, \tag{7.13}$$

where A_j is the set of nodes that start from x_j. Equation (7.12) is actually the random walk problem we have considered in Section 6.2.4. Note that the above process is conducted for every word w in the lexicon off-line, so for all the words w in all the segments x in the whole collection, we have the enhanced expected counts $E_a[w|x]$.

The enhanced language model θ_x^a for segment x is then obtained by integrating $E[w|x]$ in (7.3) and $E_a[w|x]$ in (7.12) as (7.14).

$$P(w|\theta_x^a) = \frac{E[w|x] + \mu E_a[w|x]}{L_x + \mu \sum_w E_a[w|x]}, \tag{7.14}$$

occur many times in a spoken segment, it results in reasonable results in the following experiments.

[3] The superscript ^{a} indicates that the language models are enhanced by acoustic similarity.

in which $E[w|x]$ and $E_a[w|x]$ are weighted summed with a parameter μ in the numerator, and the normalization in the denominator makes $P(w|\theta_x^a)$ become the probability in a language model. Equation (7.14) can be understood as taking $E_a[w|x]$ as the parameters of the Dirichlet prior distribution for estimating the language models of the spoken segments [161].

θ_x^a for all the segments x in the document d are then integrated to form θ_d^a in the same way as (7.7). θ_d^a is finally interpolated with the background language model to obtain a smoothed model $\bar{\theta}_d^a$ as (7.8), which is used in (7.1) for ranking.

7.3 Document Expansion with Probabilistic Latent Semantic Analysis

The problem for retrieving the documents semantically related to the query is that these documents do not necessarily contain the query term. Consider that if the query is "airplane", but the relevant document contains the term "aircraft" instead. The relevant document would have very small relevance score $S_L(Q, d)$ in (7.1) because the language models for the query and the document may be very different if they are directly estimated from the term occurrence counts in the query and the document. Hence, in such case the relevant document would be very hard to be retrieved. This problem can be solved to some extent by incorporating some latent semantic analysis approaches. Based on these approaches, the document with "aircraft" may be found to be belong to the topic about "vehicle", so we can expand the document with some terms related to "vehicle" (like "airplane") to complete its semantic representation.

Probabilistic Latent Semantic Analysis (PLSA) [154] is used for document expansion here, and Latent Dirichlet Allocation (LDA) [162] is another alternative. PLSA uses a set of latent topic variables, $\{Z_t, t = 1, 2, ..., T\}$, where T is the number of topics, to characterize the "term-document" co-occurrence relationships. Given all the spoken documents in the collection, PLSA training yields $P(w|Z_t)$, the probability of observing a word w in a document given the latent topic Z_t, and $P(Z_t|d)$, the mixture weight of topic Z_t for each document d in the collection. Hence, based on the PLSA model the probability of observing word w given document d can be parameterized by

$$P_{plsa}(w|d) = \sum_{t=1}^{T} P(w|Z_t)P(Z_t|d). \tag{7.15}$$

The parameters $P(w|Z_t)$ and $P(Z_t|d)$ is learned using the EM algorithm via maximizing the following objective function:

$$L = \sum_{d \in \mathcal{C}} \sum_{w} P(w|\theta_d) log P_{plsa}(w|d), \tag{7.16}$$

where θ_d can be either θ_d^l in Section 7.2.1 or θ_d^a in Section 7.2.2 here for spoken content. Equation (7.16) can be understood as searching for a set of parameters $P(w|Z_t)$ and $P(Z_t|d)$ minimizing the KL divergence between the document model and the PLSA-based term distribution in (7.15) for all the documents d in the collection $\mathcal{C}$.

To expand the document with semantically related words, here we adapt the background language model for each document based on its latent topics [155]. This is realized by interpolating the PLSA-based word distribution, $P_{plsa}(w|d)$ in (7.15), with the general background model θ_b in (7.9) to have a document dependent background model θ_b^d in (7.17).

$$P(w|\theta_b^d) = b_d P_{plsa}(w|d) + (1 - b_d)P(w|\theta_b) \tag{7.17}$$

where b_d is another document dependent interpolation weight which is defined as $\frac{L_d}{L_d+b}$, and b is a parameter. Then the document dependent background model θ_b^d is used to smooth the document model θ_d^a or θ_d^l as in (7.8). Therefore, after smoothed by θ_b^d, the probabilities of the words in d's language model highly related to the topics in the document d are increased.

There are others ways for incorporating the PLSA into the task of information retrieval. One way is to project both document and query into its latent topic space, and rank the documents according to the similarities of the documents and the query in terms of latent topic distributions [154]. However, this approach did not always obtain satisfied results in modern information retrieval [163].

7.4 Query Expansion with Query-regularized Mixture Model

Another common approach for handling the problem of term mismatch in information retrieval is query expansion which automatically adding some terms into the queries. The expanded queries enable the retrieval of additional documents that don't contain the original query terms but are semantically related to the queries. The basic idea for query expansion is to assume the top M documents in the first-pass retrieval results with the highest $S_L(Q, d)$ in (7.1) are relevant (or pseudo-relevant), and the terms frequently occurring in those pseudo-relevant documents may be suitable for query expansion. However, since not all pseudo-relevant documents are truly relevant, and even not all the words in the truly relevant documents are semantically related to the query, selecting useful terms for query expansion is not trivial.

7.4.1 Word-based Query Expansion

Here we borrow the query-regularized mixture model [157] originally proposed for text information retrieval for query expansion. This model assumes that the pseudo-relevant documents are composed of query-related terms and general terms, in which the ratio of the two are document-dependent. For example, the ratio for the query-related terms to the general ones is low in the irrelevant document taken as pseudo-relevant. However, those document-dependent ratios and which terms are query-related are actually unknown, but can be estimated from the term distributions in the pseudo-relevant documents. After the estimation, these query-related terms form a new query model θ'_Q, which is used to replace θ_Q in (7.1). This model is briefly summarized as below.

Suppose the pseudo-relevant documents are $\{d_1, ..., d_m, ..., d_M\}$, where M is the number of documents in this pseudo-relevant set. Each of them is composed of words generated by either the background language model θ_b in (7.9), or the query model θ'_Q which is going to be estimated. α_{d_m}, the probability of choosing θ'_Q for word generation in document d_m, is also unknown. It is possible to estimate θ'_Q and α_{d_m} for each pseudo-relevant document d_m by maximizing the likelihood of generating these pseudo-relevant documents in (7.18).

$$F_1(\theta'_Q, \alpha_{d_1}, ..., \alpha_{d_M}) = \prod_{m=1}^{M} \prod_{w} (\alpha_{d_m} P(w|\theta'_Q) + (1 - \alpha_{d_m}) P(w|\theta_b))^{P(w|\theta_{d_m})}. \tag{7.18}$$

In (7.18), the probability of generating the word w in document d_m is formulated as $\alpha_{d_m} P(w|\theta'_Q) + (1 - \alpha_{d_m}) P(w|\theta_b)$, and the document model θ_{d_m} can be either $\theta^l_{d_m}$ derived from the lattices in Section 7.2.1 or $\theta^a_{d_m}$ enhanced by the acoustic similarity in Section 7.2.2. However, θ'_Q maximizing (7.18) may be dominated by the main topics included

in the pseudo-relevant documents, which are not guaranteed to be query-related. To better handle this problem, θ'_Q is "regularized" by the original query model θ_Q in (7.2), and we define a function $F_2(\theta'_Q)$ as the prior for θ'_Q based on θ_Q.

$$F_2(\theta'_Q) = \prod_w P(w|\theta'_Q)^{P(w|\theta_Q)}, \tag{7.19}$$

in which the model θ'_Q closer to θ_Q will have larger values. θ'_Q and α_{d_m} are actually estimated by maximizing the following objective function:

$$F(\theta'_Q, \alpha_{d_1}, ..., \alpha_{d_M}) = F_1(\theta'_Q, \alpha_{d_1}, ..., \alpha_{d_M}) F_2(\theta'_Q)^{\lambda}, \tag{7.20}$$

where λ is a parameter controlling the influence of the prior function $F_2(\theta'_Q)$. The θ'_Q estimated via maximizing (7.20) would not be totally drifted away by the pseudo-relevant documents because the function $F_2(\theta'_Q)$ prefers the expanded query models similar to the original query model θ_Q.

Equation (7.20) is maximized by the EM algorithm as below:

- E step: For each word w in each document in $\{d_1, ...d_m, ...d_M\}$,

$$P(R|w, d_m) = \frac{\alpha_{d_m} P(w|\theta'_Q)}{\alpha_{d_m} P(w|\theta'_Q) + (1 - \alpha_{d_m}) P(w|\theta_b)} \tag{7.21}$$

- M step: For each document in $\{d_1, ...d_m, ...d_M\}$,

$$\alpha_{d_m} = \sum_w P(R|w, d_m) P(w|\theta_d) \tag{7.22}$$

For each word w,

$$P(w|\theta'_Q) = \frac{\lambda P(w|\theta_Q) + \sum_{m=1}^{M} P(w|\theta_d) P(R|w, d_m)}{\lambda + \sum_w \sum_{m=1}^{M} P(w|\theta_d) P(R|w, d_m)} \tag{7.23}$$

In (7.21), (7.22) and (7.23), θ_d can be either θ_d^l or θ_d^a.

7.4.2 Topic-enhanced Query Expansion

The above query expansion technique is based on words. Here we further extend the approach to a semantic version based on latent topics. Everything is in parallel with the query-regularized mixture model as summarized in the last subsection, but here instead of estimating a language model (or word distribution) θ'_Q, we now seek to estimate a query-related *topic distribution* ϕ_Q over the latent topics, that is, $\{P(Z_1|\phi_Q), ..., P(Z_t|\phi_Q), ..., P(Z_T|\phi_Q)\}$, where T is the number of topics. Here we assume the probabilities of observing all words given each latent topic $P(w|Z_t)$ are already available, which can be obtained from PLSA or other latent semantic analysis approaches. For each query Q, the topic distribution ϕ_Q is estimated via maximizing the objective function in (7.24).

$$F'(\phi_Q, \alpha_{d_1}, ..., \alpha_{d_M}) = F'_1(\phi_Q, \alpha_{d_1}, ..., \alpha_{d_M}) F'_2(\phi_Q)^{\lambda}. \tag{7.24}$$

The formulations of $F'_1(\phi_Q, \alpha_{d_1}, ..., \alpha_{d_M})$ and $F'_2(\phi_Q)$ in (7.24) are exactly the same as (7.18) and (7.19) respectively, except that $P(w|\theta'_Q)$ in (7.18) and (7.19) is replaced by $\sum_{t=1}^{T} P(w|Z_t)P(Z_t|\phi_Q)$.

Equation (7.24) is also solved by EM algorithm as below:

- E step: For each word w in each document in $\{d_1, ...d_m, ...d_M\}$,

$$P(R|w, d_m) = \frac{\alpha_{d_m} \sum_{t=1}^{T} P(w|Z_t)P(Z_t|\phi_Q)}{\alpha_{d_m} \sum_{t=1}^{T} P(w|Z_t)P(Z_t|\phi_Q) + (1 - \alpha_{d_m})P(w|\theta_b)}. \tag{7.25}$$

  For each latent topic Z_t ($t = 1$ to T),

$$P(Z_t|w) = \frac{P(w|Z_t)P(Z_t|\phi_Q)}{\sum_{t=1}^{T} P(w|Z_t)P(Z_t|\phi_Q)} \tag{7.26}$$

- M step: For each document in $\{d_1, ...d_m, ...d_M\}$,

$$\alpha_{d_m} = \sum_{w} P(R|w, d_m)P(w|\theta_d) \tag{7.27}$$

For each latent topic Z_t ($t = 1$ to T),

$$P(Z_t|\phi_Q) = \frac{\sum_w \lambda P(Z_t|w)P(w|\theta_Q) + \sum_w \sum_{m=1}^{M} P(Z_t|w)P(w|\theta_d)P(R|w,d_m)}{\lambda + \sum_w \sum_{m=1}^{M} P(w|\theta_d)P(R|w,d_m)} \tag{7.28}$$

With the semantically expanded query model ϕ_Q derived above, we have a topic-enhanced query model θ''_Q:

$$P(w|\theta''_Q) = \delta' P(w|\theta'_Q) + (1-\delta') \sum_{t=1}^{T} P(w|Z_t)P(Z_t|\phi_Q), \tag{7.29}$$

where δ' is an interpolation weight. In this way, the words semantically related to the query but not appearing in the top M documents can still be included into the query model. The topic-enhanced model θ''_Q is then used to replace θ_Q in (7.1).

7.5 Experimental Setup

In the experiments, we used the broadcast news corpus in Mandarin Chinese as the spoken document archive to be retrieved from. The news stories were recorded from radio or TV stations in Taipei from 2001 to 2003. There were a total of 5047 news stories, with a total length of 198 hours. The story length ranged from 68 to 2934 characters, with an average of 411 characters per story. 163 queries and their relevant spoken documents were provided by 22 graduate students. The number of desired documents for each query ranged from 1 to 50 with an average of 19.5, and the query length ranged from 1 to 4 Chinese words with an average of 1.6 words, or 1 to 8 Chinese characters with an average of 2.7 characters. In order to evaluate the retrieval performance of the proposed approaches with respect to different recognition conditions, we used different acoustic

and language models to transcribe the spoken documents. The two different recognition conditions Archives (A) and (B) in Section 6.4.1 were tested here as well.

For the graph construction in Section 7.2.2, nodes x_i and x_j are connected if x_i is among the K'-nearest neighbors of x_j based on $A(x_i, x_j; w)$, and x_j is among the K'-nearest neighbors of x_i, and $K' = 10$ in the experiments. The acoustic vectors used for recognition were also used to compute the acoustic similarity. Only the words occurring in the 1-best transcriptions were processed by the proposed approaches, so only 35K and 39K words were enhanced for Archives (A) and (B) respectively. For the words without enhancing, we simply set $E_a[w|x]$ equal to $E[w|x]$. Although we did not completely enhance all the words in the lexicon, the encouraging results has already been observed. Mean average precision (MAP) was used as the retrieval performance measure, and pair-wise t-test with significance level at 0.05 was used to test the significance for the performance improvement.

7.6 Experimental Results

7.6.1 Basic Language Modelling Retrieval Approach

Table 7.1 reports the results for the basic language modelling retrieval approach. The parameter a for a_d in (7.9) was set to be 1000. Rows (a) and (b) are the results for the two sets of lattices transcribed under different recognition conditions. The four columns correspond to the results using different document models θ_d in (7.1). Columns (1) and (2) are respectively the results based on the manual and 1-best transcriptions. That is, the document language models used in (7.1) was estimated based on the word occurrence counts

Table 7.1: MAP performance yielded by basic language modelling retrieval approach. The four columns correspond to the results based on manual transcriptions, 1-best transcriptions, lattices and acoustic similarity enhancement respectively. The two rows are for different recognition conditions. The superscript labels * and † respectively indicate significantly better than the results based on 1-best transcriptions and lattices.

MAP	(1) **Manual**	(2) **1-best**	(3) **Lattice**	(4) **Enhanced**
(a) Archive (A)	0.6216	0.4519	0.4579*	0.4706*†
(b) Archive (B)	0.6216	0.4956	0.5045*	0.5171*†

in the transcriptions, and smoothed by a background model trained on the transcriptions of all the spoken documents. The results based on the manual transcriptions are served as upper bound for the proposed approach [4]. Column (3) is the results that the lattice-derived document models $\bar{\theta}_d^l$ obtained in Section 7.2.1 were taken as θ_d in (7.1), while column (4) is for the acoustic similarity enhanced document models $\bar{\theta}_d^a$ with $\mu = 10$ in (7.14). Here the query model θ_Q was estimated as (7.2) without expansion. The superscript labels * and † respectively indicate significantly better than the results based on 1-best transcriptions (column (3)) and lattices (column (4)). Comparing the results in columns (1) and (2), we found that the recognition errors seriously degraded the retrieval performance. Clearly, the lattice-derived document models were better than the ones based on 1-best transcriptions (column (3) $>$ column (2)), and the proposed approach further outperformed the results merely based on the lattices (column (4) $>$ column (3)).

[4]The results for manual transcriptions were independent to the recognition conditions.

Table 7.2: KL divergence between the smoothed document models based on manual transcriptions, and 1-best transcriptions, lattices or acoustic similarity enhancement.

MAP	(1) **1-best**	(2) **Lattice**	(3) **Enhanced**
(a) Archive (A)	0.3922	0.3860	0.3748
(b) Archive (B)	0.3603	0.3538	0.3453

Table 7.2 reports the average KL divergence between the smoothed document models estimated based on the manual transcriptions, and those based on the 1-best transcriptions (column (1)), lattices (column (2)) or acoustic similarity (column (3)). We found that the smoothed language model based on acoustic similarity ($\bar{\theta}_d^a$) has the smallest KL divergence with respect to the models based on the manual transcriptions. This explains why the proposed approach improved the results merely based on the lattices in Table 7.1.

7.6.2 Document Expansion

Table 7.3 shows the results for document expansion. The parameter b for b_d in (7.17) was set to 1000. Parts (a) and (b) are respectively for different recognition conditions. The results for basic language modelling approach are taken as the baselines, which have been reported in Table 7.1. T refers to the number of topics for the PLSA models used in document expansion. Columns **Lattice** are the results based on the lattice-derived document models, that is, θ_d^l was used for PLSA training in (7.16), and used to interpolate with the document dependent background model as well. Columns **Enhanced** are the results with acoustic similarity, for which θ_d^a were used for PLSA training and interpolated with the document dependent background model. The superscript labels * and † respectively

Table 7.3: MAP performance yielded by document expansion. The results for basic language modelling approach are taken as the baselines. T is the number of topics for PLSA models. Columns **Lattice** are the results totally based on the lattice-derived document models, that is, θ_d^l was used for PLSA training in (7.16), and used to interpolate with the document dependent background model as well. Columns **Enhanced** are the results totally based on the acoustic enhanced models, for which θ_d^a were used for PLSA training and interpolated with the document dependent background model. The superscript labels * and † respectively indicate significantly better than the results of the baselines and the results based on lattices.

		(a) Archive (A)		(b) Archive (B)	
		Lattice	**Enhanced**	**Lattice**	**Enhanced**
Baseline		0.4579	$0.4706^\dagger$	0.5045	$0.5171^\dagger$
No. of	T=32	0.4855^*	$0.4936^{*\dagger}$	0.5311^*	$0.5402^{*\dagger}$
PLSA	T=64	0.4912^*	$0.5018^{*\dagger}$	0.5296^*	$0.5391^{*\dagger}$
Topics	T=128	0.4860^*	$0.4930^{*\dagger}$	0.5188^*	$0.5313^{*\dagger}$

indicate significantly better than the results of the baselines (row Baseline) and the results based on lattices (column **Lattice**). It is clear that the PLSA-based document expansion improved the retrieval performance, and the proposed enhanced model can offer extra improvements in all the conditions.

7.6.3 Query Expansion

Table 7.4 shows the results of word-based query expansion introduced in Section 7.4.1. λ in (7.20) was set to 10 in Table 7.4. Parts (a) and (b) are for two recognition conditions respectively. The results for basic language modelling approach are taken as the baselines, and they were considered as the first-pass results for selecting pseudo-relevant documents. M is the number of pseudo-relevant documents. Columns **Lattice** are for the results merely based on the lattices, that is, $\bar{\theta}_d^l$ is used for generating the first-pass results, and the expanded query model θ'_Q was estimated based on θ_d^l (θ_d in (7.18) is θ_d^l). Columns **Enhanced** are the results with acoustic similarity enhancement, or $\bar{\theta}_d^a$ for first-pass results and θ_d^a for estimating θ'_Q. We found that word-based query expansion outperformed the baselines regardless of M, and clearly acoustic similarity improved the performances in all the cases.

Table 7.5 shows the results for word-based and topic-enhanced query expansion with and without document expansion respectively. M was 10 for query expansion, and δ' in (7.29) was 0.8. The upper and lower sections in the table are respectively for different recognition conditions. Part (a) in each section is the results without document expansion, and part (b) is the results with document expansion (PLSA topic number T was 64). Row (1) is the results for word-based query expansion, or θ'_Q in Section 7.4.1 was used in (7.1). Row (2) is the results for topic-enhanced query expansion, or θ''_Q in Section 7.4.2 was used. Columns **Lattice** are the results merely based on the lattices, that is, all the operations were based on $\bar{\theta}_d^l$ and θ_d^l; while column **Enhanced** are for the results with acoustic similarity. The superscript labels † indicate significantly better than the results based on lattices. The superscript labels * indicate the topic-enhanced results sig-

nificantly better than the corresponding word-based ones, and ‡ indicate the results with document expansion significantly better than their correspondents without document expansion. Comparing the results in rows (1) and (2), the topic-enhanced query expansion further improved the word-based version (row (2) $>$ row(1)). Moreover, we found that the results in Part (b) were always better than their correspondents in Part (a). This shows that document expansion is additive with query expansion. Last but not least, the results of column **Enhanced** were always better than the results of column **Lattice** in the same row. This verified that acoustic similarity is helpful for the semantic retrieval techniques tested here.

7.7 Summary

In this chapter, we propose to enhance the expected term frequencies derived from the lattices by acoustic similarity with the graph-based approach. The enhanced term frequencies were applied on language modelling retrieval approach, document expansion and query expansion. Improved performance was observed on a corpus of broadcast news in Mandarin Chinese under different recognition conditions.

Table 7.4: MAP performance yielded by the word-based query expansion in Section 7.4.1 with $\lambda = 10$. The results for basic language modelling approach are taken as the baselines, and considered as the first-pass results for selecting pseudo-relevant documents. M is the number of pseudo-relevant documents. The superscript labels * and † respectively indicate significantly better than the results of the baselines and the results based on lattices.

	(a) Archive (A)		(b) Archive (B)	
	Lattice	**Enhanced**	**Lattice**	**Enhanced**
Baseline	0.4579	$0.4706^\dagger$	0.5045	$0.5171^\dagger$
M=5	0.4604	$0.4743^{*\dagger}$	0.5072	$0.5169^\dagger$
M=10	0.4645^*	$0.4757^{*\dagger}$	0.5116^*	$0.5206^\dagger$
M=15	0.4657^*	$0.4789^{*\dagger}$	0.5156^*	$0.5262^{*\dagger}$
M=20	0.4652^*	$0.4792^{*\dagger}$	0.5156^*	$0.5266^{*\dagger}$
M=25	0.4671^*	$0.4803^{*\dagger}$	0.5144^*	$0.5293^{*\dagger}$
M=30	0.4673^*	$0.4811^{*\dagger}$	0.5141^*	$0.5273^{*\dagger}$
M=35	0.4675^*	$0.4816^{*\dagger}$	0.5127^*	$0.5295^{*\dagger}$
M=40	0.4673^*	$0.4813^{*\dagger}$	0.5123^*	$0.5290^{*\dagger}$
M=45	0.4661^*	$0.4815^{*\dagger}$	0.5103	$0.5270^{*\dagger}$
M=50	0.4672^*	$0.4818^{*\dagger}$	0.5082	$0.5267^{*\dagger}$

Table 7.5: MAP performance for word-based and topic-enhanced query expansion with and without document expansion. $\lambda = 10$ for query expansion. The superscript labels † indicate significantly better than the results based on lattices. The superscript labels * indicate the topic-enhanced results significantly better than the corresponding word-based ones, and ‡ indicate the results with document expansion in part (b) significantly better than their correspondents without document expansion in part (a).

Recognition Conditions	Document Expansion	Query Expansion	**Lattice**	**Enhanced**
Archive (A)	(a) **NO**	(1) word	0.4645	$0.4757^{\dagger}$
		(2) topic	0.4693^{*}	$0.4799^{\dagger *}$
	(b) **YES**	(1) word	$0.4965^{\ddagger}$	$0.5048^{\dagger\ddagger}$
		(2) topic	$0.4976^{\ddagger}$	$0.5069^{\dagger\ddagger}$
Archive (B)	(a) **NO**	(1) word	0.5116	$0.5206^{\dagger}$
		(2) topic	0.5159^{*}	$0.5231^{\dagger}$
	(b) **YES**	(1) word	$0.5333^{\ddagger}$	$0.5421^{\dagger\ddagger}$
		(2) topic	$0.5350^{\ddagger}$	$0.5436^{\dagger\ddagger}$

Chapter 8 Conclusion and Future Work

8.1 Conclusion

About four years ago (2009), when I started to research for spoken content retrieval, the cascade of the recognition system and text-based retrieval system has achieved many successful results. People have found that lattices can offer extra benefit over the 1-best transcriptions, and many efficient approaches for retrieving such lattice structures have been investigated. The problem of the OOV queries was addressed by subword-based indexing and grapheme-to-phoneme techniques. Although the poor recognition usually degrades the retrieval performance, this problem would be solved by the researchers studying acoustic and language models one day, and does not have too much relation with the ones researching spoken content retrieval. At that time, people did not aware the possibilities of coupling the recognition and retrieval system and considering the information beyond recognition output.

Today (2012) several novel methods aiming at breaking the boundary between recognition and retrieval are already proposed, which are summarized in this thesis. The acoustic models can be re-estimated by user relevance feedback taking into account the nature of the retrieval task in Chapter 4. Machine learning methods can take the acoustic vectors such as MFCC as features, and are successfully applied on pseudo-relevance feedback (PRF) in Chapter 5. Moreover, in Chapter 6, the acoustic similarity is able to compensate for the information lost in the recognition stage, which is used in PRF and a graph-based approach. Those approaches were all verified on lecture courses and broadcast news, and the example-based approaches can further improve subword-based retrieval system and

thus improve the performance of a set of OOV queries. In this thesis, semantic retrieval is also considered, in which the acoustic similarity is used to enhance the term frequency estimation by a graph-based approach.

I believed that breaking the boundary between recognition and retrieval is the future trend for spoken content retrieval. This thesis just opens the doors to these kinds of ideas, and lots of related research topics are waited to be explored and investigated. More experiments and analysis for more different corpora are certainly necessary, and more fancy algorithms and powerful methods can be expected in the near future.

8.2 Future Work

It is necessary to test the proposed approaches on some benchmark corpora. I already conducted the SVM-based method in Chapter 5 and example-based approach in Chapter 6 on the WSJ0 SI-84 (or the training set for the Aurora-4), and the improvements were observed in the preliminary experiments. Although I do not report these experiments here due to time limitation, I hope these results can be published in the near future.

The tremendous amount of parameters in the acoustic models and very limited training data from relevance feedback make the acoustic model re-estimation process risky. Instead of re-estimating all the parameters in the acoustic models, for the future work it is possible to estimate a linear transform for the means of the HMM for each phone class just like MLLR, or estimate the transform of feature dimension reduction like linear discriminant analysis (LDA). It is also possible to estimate language model parameters by relevance feedback. However, re-estimating the n-gram probabilities did not achieve any improvements in some preliminary experiments. I found that the information from rele-

vance feedback is too sparse for estimating the n-gram probabilities, so the probabilities re-estimated by those labelled spoken segments seldom influence the scores of unlabelled segments. Those results have not been reported yet. For language model re-estimation, estimating the class transition probabilities for the class-based langauge models may be more realistic. Recent developed continuous language models are another alternatives. Since these models project the word onto a feature space, and let the different n-gram probabilities share the common parameters, these models may address the problem of training data sparsity.

For applying the machine learning methods for spoken content retrieval, certainly lots of approaches beyond SVM should be tested and investigated. There are several problems to be solved:

1. The positions of the query hypotheses may be inaccurate, and sometimes there can be several candidate hypothesis regions in a spoken segment.

2. The state boundaries may be inaccurate as well.

Solving the above problems may especially improve the performance of the OOV queries, since their hypothesis regions in the segments are usually unclear.

For the example-based approach, only the simple DTW-based approach is used to compare the two acoustic vector sequences in this thesis. In the future work, any state-of-the-art approach can be applied to improve the performance. Moreover, there are actually several graph-based approaches in the literatures. I have tested the associated network which makes the scores of the nodes connected become locally smooth via solving a quadratic programming problem, but it did not outperform the random walk.

For the term frequency estimation enhancement, actually any state-of-the-art STD

methods can be applied here, and the enhanced term frequencies can be applied on any task about spoken language understanding, for example, speech summarization. On the other hand, semantic analysis and STD can be reinforced. The term frequencies enhanced by STD improve the performance of semantic analysis; while semantic analysis can verify the correctness for some terms in the spoken content, and this information can further enhance the STD model.

Bibliography

[1] Lin-Shan Lee and Berlin Chen, "Spoken document understanding and organization," *Signal Processing Magazine, IEEE*, vol. 22, pp. 42 – 60, 2005.

[2] C. Chelba, T.J. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 39 –49, may 2008.

[3] John S. Garofolo, Cedric G. P. Auzanne, and Ellen M. Voorhees, *The TREC Spoken Document Retrieval Track: A Success Story*, 2000.

[4] Murat Saraclar, "Lattice-based search for spoken utterance retrieval," in *In Proceedings of HLT-NAACL 2004*, 2004, pp. 129–136.

[5] Gokhan Tur and Renato DeMori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, chapter 15, pp. 417–446, John Wiley & Sons Inc, 2011.

[6] Lin-Shan Lee and Yi-Cheng Pan, "Voice-based information retrieval: how far are we from the text-based information retrieval ?," in *ASRU*, 13 2009-dec. 1 2009, pp. 26 –43.

[7] David R. H. Miller, Michael Kleber, Chia lin Kao, and Owen Kimball, "Rapid and accurate spoken term detection," in *INTERSPEECH*, 2007.

[8] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *INTERSPEECH*, 2007.

[9] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 615–622.

[10] Peng Yu, Kaijiang Chen, Lie Lu, and Frank Seide, "Searching the audio notebook: keyword search in recorded conversations," in *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 947–954.

[11] Ciprian Chelba and Alex Acero, "Position specific posterior lattices for indexing speech," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 443–450.

[12] S. Parlak and M. Saraclar, "Spoken term detection for Turkish broadcast news," in *ICASSP*, 2008.

[13] Takaaki Hori, I. Lee Hetherington, Timothy J. Hazen, and James R. Glass, "Open-vocabulary spoken utterance retrieval using confusion networks," in *ICASSP*, 2007.

[14] Jonathan Mamou, David Carmel, and Ron Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, SIGIR '06, pp. 51–58.

[15] Jorge Silva, Ciprian Chelba, and Alex Acero, "Pruning analysis for the position specific posterior lattices for spoken document search," in *ICASSP*, 2006.

[16] Jorge Silva, Ciprian Chelba, and Alex Acero, "Integration of metadata in spoken document search using position specific posterior latices," in *SLT*, 2006.

[17] Zheng-Yu Zhou, Peng Yu, Ciprian Chelba, and Frank Seide, "Towards spoken-document retrieval for the internet: lattice indexing for large-scale web-search architectures," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006, pp. 415–422.

[18] Frank Seide, Peng Yu, and Yu Shi, "Towards spoken-document retrieval for the enterprise: Approximate word-lattice indexing with text indexers," in *ASRU*, 2007.

[19] Yi-Cheng Pan, Hung-Lin Chang, and Lin-Shan Lee, "Analytical comparison between position specific posterior lattices and confusion networks based onwords and subword units for spoken document indexing," in *ASRU*, 2007.

[20] Cyril Allauzen, Mehryar Mohri, and Murat Saraclar, "General indexation of weighted automata: application to spoken utterance retrieval," in *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, 2004.

[21] B. Logan, P. Moreno, J. M. Van Thong, and E. Whittacker, "An experimental study of an audio indexing system for the web," in *ICSLP*, 2000.

[22] Yi-Cheng Pan, Hung-Lin Chang, , and Lin-Shan Lee, "Subword-based position specific posterior lattices (S-PSPL) for indexing speech information," in *INTERSPEECH*, 2007.

[23] Sha Meng, Peng Yu, Frank Seide, and Jia Liu, "A study of lattice-based spoken term detection for chinese spontaneous speech," in *ASRU*, 2007.

[24] J. Scott Olsson, Jonathan Wintrode, and Matthew Lee, "Fast unconstrained audio search in numerous human languages," in *ICASSP*, 2008.

[25] Corentin Dubois and Delphine Charlet, "Using textual information from LVCSR transcripts for phonetic-based spoken term detection," in *ICASSP*, 2008.

[26] Dogan Can, Erica Cooper, Abhinav Sethy, Chris White, Bhuvana Ramabhadran, and Murat Saraclar, "Effect of pronunciations on OOV queries in spoken term detection," in *ICASSP*, 2009.

[27] Dong Wang, Simon King, and Joe Frankel, "Stochastic pronunciation modelling for spoken term detection," in *INTERSPEECH*, 2009.

[28] Roy Wallace, Robbie Vogt, and Sridha Sridharan, "A phonetic search approach to the 2006 NIST spoken term detection evaluation," in *INTERSPEECH*, 2007.

[29] Yoshiaki Itoh, Kohei Iwata, Kazunori Kojima, Masaaki Ishigame, Kazuyo Tanaka, and Shi wook Lee, "An integration method of retrieval results using plural subword models for vocabulary-free spoken document retrieval," in *INTERSPEECH*, 2007.

[30] K. Ng, *Subword-based approaches for spoken document retrieval*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[31] B. Logan, J.-M. Van Thong, and P.J. Moreno, "Approaches to reduce the effects of OOV queries on indexed spoken audio," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 899 – 906, oct. 2005.

[32] Ville T. Turunen, "Reducing the effect of OOV query words by using morph-based spoken document retrieval," in *INTERSPEECH*, 2008.

[33] Ville T. Turunen and Mikko Kurimo, "Indexing confusion networks for morph-based spoken document retrieval," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, SIGIR '07, pp. 631–638.

[34] Dong Wang, Joe Frankel, Javier Tejedor, and Simon King, "A comparison of phone and grapheme-based spoken term detection," in *ICASSP*, 2008.

[35] Shi wook Lee, Kazuyo Tanaka, and Yoshiaki Itoh, "Combining multiple subword representations for open-vocabulary spoken document retrieval," in *ICASSP*, 2005.

[36] Sha Meng, Peng Yu, Jia Liu, and Frank Seide, "Fusing multiple systems into a compact lattice index for chinese spoken term detection," in *ICASSP*, 2008.

[37] Chao-Hong Meng, Hung-Yi Lee, and Lin-Shan Lee, "Improved lattice-based spoken document retrieval by directly learning from the evaluation measures," in *ICASSP*, 2009.

[38] Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *ASRU*, 2009.

[39] Timothy J. Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *ASRU*, 2009.

[40] Yaodong Zhang and James R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on gaussian posteriorgrams," in *ASRU*, 2009.

[41] W. Shen, C. White, and T. Hazen, "A comparison of query-by-example methods for spoken term detection," in *INTERSPEECH*, 2009.

[42] Hui Lin, Alex Stupakov, and Jeff Bilmes, "Improving multi-lattice alignment based spoken keyword spotting," in *ICASSP*, 2009.

[43] Hui Lin, Alex Stupakov, and Jeff Bilmes, "Spoken keyword spotting via multi-lattice alignment," in *INTERSPEECH*, 2008.

[44] Chun-An Chan and Lin-Shan Lee, "Unsupervised spoken term detection with spoken queries using segment-based dynamic time warping," in *INTERSPEECH*, 2010.

[45] Chun-An Chan and Lin-Shan Lee, "Unsupervised hidden markov modeling of spoken queries for spoken term detection without speech recognition," in *INTERSPEECH*, 2011.

[46] Haipeng Wang, Cheung-Chi Leung, Tan Lee, Bin Ma, and Haizhou Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *ICASSP*, 2012.

[47] `http://www.itl.nist.gov/iad/mig/tests/std/2006/index.html`.

[48] *Text REtrieval Conference,* `http://trec.nist.gov/`.

[49] Hung-Yi Lee and Lin-Shan Lee, "Improving retrieval performance by user feedback: a new framework for spoken term detection," in *ICASSP*, 2010.

[50] Yu-Hui Chen, Chia-Chen Chou, Hung-Yi Lee, and Lin-Shan Lee, "An initial attempt to improve spoken term detection by learning optimal weights for different indexing features," in *ICASSP*, 2010, pp. 5278 –5281.

[51] Roy Wallace, Robbie Vogt, Brendan Baker, and Sridha Sridharan, "Optimising figure of merit for phonetic spoken term detection," in *ICASSP*, 2010.

[52] Javier Tejedor, Doroteo T. Toledano, Miguel Bautista, Simon King, Dong Wang, and Jose Colas, "Augmented set of features for confidence estimation in spoken term detection," in *INTERSPEECH*, 2010.

[53] Dong Wang, Simon King, Joe Frankel, and Peter Bell, "Term-dependent confidence for out-of-vocabulary term detection," in *INTERSPEECH*, 2009.

[54] Joseph Keshet, David Grangier, and Samy Bengio, "Discriminative keyword spotting," *Speech Communication*, vol. 51, pp. 317 – 329, 2009.

[55] M. Wollmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional lstm networks," in *ICASSP*, 2009.

[56] Tomoyosi Akiba, Hiromitsu Nishizaki, Kiyoaki Aikawa, Tatsuya Kawahara, and Tomoko Matsui, "Overview of the IR for spoken documents task in NTCIR-9 workshop," in *Proceedings of NTCIR-9 Workshop*, 2011.

[57] J.-M. Van Thong, P.J. Moreno, B. Logan, B. Fidler, K. Maffey, and M. Moores, "Speechbot: an experimental speech-based search engine for multimedia content

on the web," *Multimedia, IEEE Transactions on*, vol. 4, no. 1, pp. 88 –96, mar 2002.

[58] *http://speechfind.utdallas.edu/*.

[59] *http://www.ngsw.org/*.

[60] Masataka Goto, Jun Ogata, and Kouichirou Eto, "Podcastle: A web 2.0 approach to speech recognition research," in *INTERSPEECH*, 2007.

[61] Jun Ogata and Masataka Goto, "Podcastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription," in *INTERSPEECH*, 2009.

[62] Christopher Alberti, Michiel Bacchiani, Ari Bezman, Ciprian Chelba, Anastassia Drofa, Hank Liao, Pedro Moreno, Ted Power, Arnaud Sahuguet, Maria Shugrina, and Olivier Siohan, "An audio indexing system for election video material," in *ICASSP*, 2009.

[63] *labs.google.com/gaudi*.

[64] *http://web.sls.csail.mit.edu/lectures/*.

[65] Bo-June Hsu and J. Glass, "Language model parameter estimation using user transcriptions," in *ICASSP*, 2009.

[66] Gregory T Yu, "Efficient error correction for speech systems using constrained re-recognition," M.S. thesis, Massachusetts Institute of Technology, 2008.

[67] Hung-Yi Lee, Yueh-Lien Tang, Hao Tang, and Lin-Shan Lee, "Spoken term detection from bilingual spontaneous speech using code-switched lattice-based structures for words and subword units," in *ASRU*, 2009.

[68] Sheng-Yi Kong, Miao-Ru Wu, Che-Kuang Lin, Yi-Sheng Fu, and Lin-Shan Lee, "Learning on demand – course lecture distillation by information extraction and semantic structuring for spoken documents," in *ICASSP*, 2009.

[69] I. Ruthven and M. Lalmas, "A survey on the use of relevance feedback for information access systems," in *The Knowledge Engineering Review*, 2003.

[70] J. J. Rocchio, "Relevance feedback in information retrieval," in *The SMART retrieval system - experiments in automatic document processing*, 1971.

[71] S. E. Robertson and K. Sparck Jones, "Relevance weighting of search terms," in *Journal of the American Society for Information Science*, 1976.

[72] Chengxiang Zhai and John Lafferty, "Model-based feedback in the language modeling approach to information retrieval," in *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, 2001, pp. 403–410.

[73] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," in *Multimedia systems*, 2003.

[74] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, pp. 5:1–5:60, May 2008.

[75] Pengyu Hong, Qi Tian, and T.S. Huang, "Incorporate support vector machines to content-based image retrieval with relevance feedback," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2000.

[76] S.D. MacArthur, C.E. Brodley, and Chi-Ren Shyu, "Relevance feedback decision trees in content-based image retrieval," in *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, 2000.

[77] Jing Xin and J.S. Jin, "Learning from user feedback for image retrieval," in *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, 2003.

[78] N. Vasconcelos and A. Lippman, "Bayesian relevance feedback for content-based image retrieval," in *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, 2000.

[79] Rong Yan, Alexander Hauptmann, and Rong Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proceedings of the eleventh ACM international conference on Multimedia*, 2003.

[80] Rong Yan, Alexander Hauptmann, and Rong Jin, "Multimedia search with pseudo-relevance feedback," in *Proceedings of the 2nd international conference on Image and video retrieval*, 2003, CIVR'03, pp. 238–247.

[81] Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Mingjing Li, "Online video recommendation based on multimodal fusion and relevance

feedback," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007.

[82] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan, "Optimizing web search using web click-through data," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 2004, CIKM '04, pp. 118–126.

[83] Olivier Chapelle and Ya Zhang, "A dynamic bayesian network click model for web search ranking," in *Proceedings of the 18th international conference on World wide web*, 2009, WWW '09, pp. 1–10.

[84] Diane Kelly and Jaime Teevan, "Implicit feedback for inferring user preference: a bibliography," *SIGIR Forum*, vol. 37, no. 2, pp. 18–28, Sept. 2003.

[85] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, SIGIR '05, pp. 154–161.

[86] Diane Kelly and Nicholas J. Belkin, "Display time as implicit feedback: understanding task effects," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004.

[87] Xuehua Shen, Bin Tan, and ChengXiang Zhai, "Context-sensitive information retrieval using implicit feedback," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, SIGIR '05, pp. 43–50.

[88] Georg Buscher, Andreas Dengel, and Ludger van Elst, "Eye movements as implicit relevance feedback," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, 2008, pp. 2991–2996.

[89] Georg Buscher, Andreas Dengel, and Ludger van Elst, "Query expansion using gaze-based feedback on the subdocument level," in *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 387–394.

[90] Jarkko Salojarvi, Kai Puolamaki, and Samuel Kaski, "Implicit relevance feedback from eye movements," in *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, 2005, pp. 513–518.

[91] Ioannis Arapakis, Joemon M. Jose, and Philip D. Gray, "Affective feedback: an investigation into the role of emotions in the information seeking process," in *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 395–402.

[92] K. Bain, S. Basson, A. Faisman, and D. Kanevsky, "Accessibility, transcription, and access everywhere," *IBM Systems Journal*, vol. 44, pp. 589 –603, 2005.

[93] Thorsten Joachims and Filip Radlinski, "Search engines that learn from implicit feedback," *Computer*, vol. 40, pp. 34–40, 2007.

[94] T. Deselaers, R. Paredes, E. Vidal, and H. Ney, "Learning weighted distances for relevance feedback in image retrieval," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008.

[95] Donna Harman, "Relevance feedback revisited," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, 1992.

[96] Makoto Iwayama, "Relevance feedback with a small number of relevance judgements: incremental relevance feedback vs. document clustering," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

[97] H. Muller, W. Muller, S. Marchand-Maillet, T. Pun, and D.M. Squire, "Strategies for positive and negative relevance feedback in image retrieval," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000.

[98] Chris Buckley, Gerard Salton, and James Allan, "The effect of adding relevance information in a relevance feedback environment," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994.

[99] Oren Kurland, Lillian Lee, and Carmel Domshlak, "Better than the real thing?: iterative pseudo-query processing using cluster-based language models," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005.

[100] Jinxi Xu and W. Bruce Croft, "Query expansion using local and global document analysis," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996.

[101] Victor Lavrenko and W. Bruce Croft, "Relevance based language models," in *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 120–127.

[102] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma, "Improving pseudo-relevance feedback in web information retrieval using web page segmentation," in *Proceedings of the 12th international conference on World Wide Web*, 2003.

[103] Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama, "Flexible pseudo-relevance feedback via selective sampling," *ACM Transactions on Asian Language Information Processing*, vol. 4, pp. 111–135, 2005.

[104] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[105] Kyung Soon Lee, W. Bruce Croft, and James Allan, "A cluster-based resampling method for pseudo-relevance feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[106] Yuanhua Lv and ChengXiang Zhai, "A comparative study of methods for estimating query language models with pseudo feedback," in *Proceeding of the 18th ACM conference on Information and knowledge management*, 2009.

[107] Yuanhua Lv and ChengXiang Zhai, "Positional relevance model for pseudo-relevance feedback," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[108] W.-H. Lin, R. Jin, and A. Hauptmann, "Web image retrieval re-ranking with relevance model," in *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, 2003, pp. 242 – 248.

[109] S. Rudinac, M. Larson, and A. Hanjalic, "Exploiting visual reranking to improve pseudo-relevance feedback for spoken-content-based video retrieval," in *Image Analysis for Multimedia Interactive Services, 2009. WIAMIS '09. 10th Workshop on*, 2009.

[110] C. Parada, A. Sethy, and B. Ramabhadran, "Balancing false alarms and hits in spoken term detection," in *ICASSP*, 2010.

[111] Savitha Srinivasan and Dragutin Petkovic, "Phonetic confusion matrix based spoken document retrieval," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, SIGIR '00, pp. 81–87.

[112] H. Nanjo and T. Kawahara, "A new ASR evaluation measure and minimum bayes-risk decoding for open-domain speech understanding," in *ICASSP*, 2005.

[113] T. Shichiri, H. Nanjo, and T. Yoshimi, "Minimum bayes-risk decoding with presumed word significance for speech based information retrieval," in *ICASSP*, 2008.

[114] Q. Fu and B.-H. Juang, "Automatic speech recognition based on weighted minimum classification error (W-MCE) training method," in *ASRU*, 2007.

[115] Biing-Hwang Juang, Wu Hou, and Chin-Hui Lee, "Minimum classification error rate methods for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 3, pp. 257 –265, may 1997.

[116] J. Shao, R.-P. Yu, Q. Zhao, Y. Yan, and F. Seide, "Towards vocabulary-independent speech indexing for large-scale repositories," in *INTERSPEECH*, 2008.

[117] Hung-Yi Lee, Chia-Ping Chen, Ching-Feng Yeh, and Lin-Shan Lee, "Improved spoken term detection by discriminative training of acoustic models based on user relevance feedback," in *INTERSPEECH*, 2010.

[118] Hung-Yi Lee, Chia-Ping Chen, Ching-Feng Yeh, and Lin-Shan Lee, "A framework integrating different relevance feedback scenarios and approaches for spoken term detection," in *SLT*, 2012.

[119] Hung-Yi Lee, Chia-Ping Chen, and Lin-Shan Lee, "Integrating recognition and retrieval with relevance feedback for spoken term detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 7, pp. 2095 –2110, sept. 2012.

[120] Jen-Tzung Chien and Meng-Sung Wu, "Minimum rank error language modeling," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, pp. 267 –276, 2009.

[121] Daniel Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Dept, 2003.

[122] B.-Y. Liang, "Acoustic models for continuous mandarin speech recognition," M.S. thesis, NTU, 1998.

[123] Tsung-Wei Tu, Hung-Yi Lee, and Lin-Shan Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," in *ASRU*, 2011.

[124] Hung-Yi Lee, Tsung-Wei Tu, Chia-Ping Chen, Chao-Yu Huang, and Lin-Shan Lee, "Improved spoken term detection using support vector machines based on lattice context consistency," in *ICASSP*, 2011.

[125] Christopher J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, June 1998.

[126] Yaodong Zhang and J.R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *ICASSP*, 2010.

[127] S.C.H. Hoi, M.R. Lyu, and R. Jin, "A unified log-based relevance feedback scheme for image retrieval," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 4, pp. 509 – 524, april 2006.

[128] Thomas G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, Aug. 2000.

[129] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims, "A support vector method for optimizing average precision," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[130] Y. Chen, X. Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *Proc. IEEE ICIP*, 2002.

[131] Xiang Sean Zhou and T.S. Huang, "Small sample learning during multimedia retrieval using biasmap," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001.

[132] Masashi Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd international conference on Machine learning*, 2006, ICML '06, pp. 905–912.

[133] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[134] Thorsten Joachims, "Transductive inference for text classification using support vector machines," in *16th International Conference on Machine Learning*, 1999.

[135] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[136] Chia-Ping Chen, Hung-Yi Lee, Ching-Feng Yeh, and Lin-Shan Lee, "Improved spoken term detection by feature space pseudo-relevance feedback," in *INTERSPEECH*, 2010.

[137] Yun-Nung Chen, Chia-Ping Chen, Hung-Yi Lee, Chun-An Chan, and Lin-Shan Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *ICASSP*, 2011.

[138] Hung-Yi Lee, Yun-Nung Chen, and Lin-Shan Lee, "Improved speech summarization and spoken term detection with graphical analysis of utterance similarities," in *APSIPA*, 2011.

[139] Hung-Yi Lee, Po-Wei Chou, and Lin-Shan Lee, "Open-vocabulary retrieval of spoken content with shorter/longer queries considering word/subword-based acoustic feature similarity," in *INTERSPEECH*, 2012.

[140] Amy N. Langville and Carl D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM Rev.*, vol. 47, pp. 135–161, January 2005.

[141] Sergey Brin and Lawrence Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107 – 117, 1998.

[142] Winston H. Hsu, Lyndon S. Kennedy, and Shih-Fu Chang, "Video search reranking through random walk over document-level context graph," in *Proceedings of the 15th international conference on Multimedia*, 2007, pp. 971–980.

[143] Xinmei Tian, Linjun Yang, Jingdong Wang, Yichen Yang, Xiuqing Wu, and Xian-Sheng Hua, "Bayesian video search reranking," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 131–140.

[144] G. Aradilla, J. Vepa, and H. Bourlard, "Using posterior-based features in template matching for speech recognition," in *ICSLP*, 2006.

[145] Ching-Feng Yeh, Liang-Che Sun, Chao-Yu Huang, and Lin-Shan Lee, "Bilingual acoustic modeling with state mapping and three-stage adaptation for transcribing unbalanced code-mixed lectures," in *ICASSP*, 2011.

[146] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, pp. 434 – 451, 2008.

[147] Ariya Rastrow, Abhinav Sethy, Bhuvana Ramabhadran, and Frederick Jelinek, "Towards using hybrid word and fragment units for vocabulary independent LVCSR systems," in *INTERSPEECH*, 2009.

[148] Tsung-Wei Tu, Hung-Yi Lee, Yu-Yu Chou, and Lin-Shan Lee, "Semantic query expansion and context-based discriminative term modeling for spoken document retrieval," in *ICASSP*, 2012.

[149] Hung-Lin Chang, Yi-Cheng Pan, and Lin-Shan Lee, "Latent semantic retrieval of spoken documents over position specific posterior lattices," in *SLT*, 2008.

[150] B. Chen, Pei-Ning Chen, and Kuan-Yu Chen, "Query modeling for spoken document retrieval," in *ASRU*, 2011.

[151] Xinhui Hu, Ryosuke Isotani, Hisashi Kawai, and Satoshi Nakamura, "Cluster-based language model for spoken document retrieval using NMF-based document clustering," in *INTERSPEECH*, 2010.

[152] Tomoyosi Akiba and Koichiro Honda, "Effects of query expansion for spoken document passage retrieval," in *INTERSPEECH*, 2011.

[153] Ryo Masumura, Seongjun Hahm, and Akinori Ito, "Language model expansion using webdata for spoken document retrieval," in *INTERSPEECH*, 2011.

[154] Thomas Hofmann, "Probabilistic latent semantic analysis," in *In Proc. of Uncertainty in Artificial Intelligence, UAI'99*, 1999.

[155] Xing Wei and W. Bruce Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, SIGIR '06, pp. 178–185.

[156] Quan Wang, Jun Xu, Hang Li, and Nick Craswell, "Regularized latent semantic indexing," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, SIGIR '11, pp. 685–694.

[157] Tao Tao and ChengXiang Zhai, "Regularized estimation of mixture models for robust pseudo-relevance feedback," in *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 162–169.

[158] Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tou Ng, "Statistical lattice-based spoken document retrieval," *ACM Trans. Inf. Syst.*, vol. 28, pp. 2:1–2:30, 2010.

[159] Chengxiang Zhai and John Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, SIGIR '01, pp. 334–342.

[160] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson, "A probabilistic model of information retrieval: Development and comparative experiments," in *Information Processing and Management*, 2000.

[161] Matthew Lease and Eugene Charniak, "A dirichlet-smoothed bigram model for retrieving spontaneous speech," in *Advances in Multilingual and Multimodal Information Retrieval*. Springer-Verlag, 2008.

[162] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[163] Avinash Atreya and Charles Elkan, "Latent semantic indexing (LSI) fails for TREC collections," *SIGKDD Explor. Newsl.*, vol. 12, pp. 5–10, 2011.