Speech Retrieval under Limited Resources and Open Domain Conditions

Justin Chiu

June 2014

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Alexander Rudnicky, Chair (Carnegie Mellon University)
Alan W Black (Carnegie Mellon University)
Alexander G. Hauptmann (Carnegie Mellon University)
Gareth J.F. Jones (Dublin City University)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Keywords: Speech Retrieval, Limited Resources, Open Domain

Abstract

Speech Retrieval focuses on retrieving a segment of speech from a speech corpus correspond to a given query. A standard Speech Retrieval system usually composed by two systems, the Automatic Speech Recognition (ASR) system and the Information Retrieval (IR) system. The ASR system transcribes the speech and represents the transcript in different formats. The transcript is then indexed and searched by the IR system. As a result, Speech Retrieval is sensitive to the ASR, since IR system is depend on the transcript generated by ASR.

The current challenge in Speech Retrieval is the limitation of ASR performance under certain conditions. Two such conditions are Limited Resources and Open Domain. Under Limited resources condition, the training data is not sufficient for creating a robust ASR system. A good example for this condition is to perform Speech Retrieval on limited resources languages such as Tagalog or Assamese. On the other hand, under Open Domain condition, the recorded speech varies in many perspectives. The high diversity of recorded speech limits the performance of a single ASR system. A good example for this condition is to perform Speech Retrieval on YouTube videos, such as online lectures.

We believe Speech Retrieval under these conditions can be significantly improved from different approaches. The first one is to apply extra information, such as contexts from conversation. A context includes the other words in the same utterance or conversation. The second approach is to refine the existing IR system, by using better IR search strategy for Speech Retrieval. We analysis existing IR system and present a better search strategy, which is based on the diversity of current approaches.

We have investigated how to integrate these two approaches to Speech Retrieval, and determined that the approaches can achieve improvement on Spoken Term Detection (STD) under the limited resources condition. The resulting system has been shown to be effective on multiple languages, implying that the improvement is language independent.

Based on our positive result regarding the limited resources condition, we propose to extend existing approaches and develop new techniques for better Speech Retrieval under the open domain condition. We propose a new Speech Retrieval task called Spoken Snippet Retrieval (SSR), which retrieve a moderate size of speech from the speech collection with just enough context. The retrieved snippet is easier for user to listen through compare to the spoken document retrieved by Spoken Document Retrieval (SDR) systems, which has average length of 3 minutes. The snippet is more comprehensible compare to the term location detected by STD systems, since the context are given. The main contribution for the thesis is to complete SSR on the open domain data, which we believe is doing the adequate retrieval on the appropriate data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first introduce Speech Retrieval. Following that, we present the current challenge in Speech Retrieval, proposal statement, a summary of completed work, proposed work and proposal organization.

1.1 Speech Retrieval

Speech Retrieval focus on retrieving a segment of speech that is related to the given query from a corpus. There are two tasks that received most research efforts in Speech Retrieval.

- Spoken Document Retrieval (SDR) [17]. SDR provides content-based retrieval of excerpts from archives of recordings of speech.
- Spoken Term Detection (STD) [16]. STD aims to detect the presence of a query word in a speech corpus.

Aside from these two tasks, Topic Detection and Summarization of speech data can also be regarded as tasks in Speech Retrieval; they retrieve essential content from huge archives of speech recordings.

Most Speech Retrieval task is accomplished by using a combination of Automatic Speech Recognition (ASR) and Information Retrieval (IR) system. An ASR system is applied to an audio stream and generates a time-marked transcription of the speech. The transcription may be phone- or word-based in either a lattice (probability network), n-best list (multiple individual transcriptions), or more typically, a 1-best transcript (the most probable transcription as determined by the recognizer). The transcript is then indexed and searched by a IR system, and the result returned for a query is a list of temporal pointers to the audio stream ordered by decreasing similarity between the content of the speech being pointed to and the query. [17]

Among these two systems, the ASR system is generally considered more important, since any correctly retrieved result is based on proper recognition. As a result, works in the recent Speech Retrieval domain are trying to improve the recognition performance, which can be estimated by reducing the Word Error Rate (WER) or increasing the lattice recall in ASR for better Speech Retrieval performance.

1.2 Current challenge in Speech Retrieval

The current challenge in Speech Retrieval is the limitation of ASR performance under certain conditions. Even with the recent advance in Deep Neural Networks for ASR, there are still conditions under which ASR performs badly. Two such conditions are:

- Limited resources: Under this condition, the training data is not sufficient for creating a robust ASR system. A good example for this condition is to perform Speech Retrieval on limited resources languages such as Tagalog or Assamese.
- Open domain: Under this condition, the recorded speech varies in many perspectives. The high diversity of recorded speech limits the performance of a single ASR system. A good example for this condition is to perform Speech Retrieval on YouTube videos, such as online lectures.

The two conditions often arise in real world applications: There are thousands of languages in existence, and a large number of them do not have enough labeled data for training a robust ASR system. The cost for collecting enough data for every language is simply too high. It is also not possible to create domain specific ASR systems for open domain speech recordings. The source of these recordings is constantly growing and changing. New speech data are collected with different recording devices, ranging from traditional audio recorders to modern smart phones. Due to the vast range of recording devices in use and varying environments in which recordings are made, creating specific ASR system for each scenario becomes an impossible task.

1.3 Proposal Statement

We believe Speech Retrieval under these conditions can be significantly improved by two different approaches: applying contexts from conversation for hypothesis rescoring and improving IR system.

The first one is to apply contexts from conversation for hypothesis rescoring. A context includes the other words in the same utterance or conversation. This approach relies on our understanding of conversational structure, and is especially helpful when the baseline system does not have robust performance. We select Speech Retrieval as the target task for applying context from conversation, since its current challenge is on the limited quality ASR result. We believe that when the ASR result is not ideal, we should introduce external knowledge that diverse with the existing ASR result to improve Speech Retrieval performance. Therefore, we apply rescoring algorithm based on context from conversation on the ASR result, and achieve better Speech Retrieval result, given the same ASR system in the Speech Retrieval pipeline.

The second approach is to improve IR systems. The IR system searches the results generated by the ASR system. We can assume each IR system generates an approximation of word occurrences according to its indexing and search algorithm. We design a better search strategy based on the combination of two existing approaches. By combining multiple approximations of word occurrences, we can have better understanding of actual occurrence of the word. This approach also relies on obtaining more diversity from the same ASR system, we believe introducing and combining diversity can bring improvement to the limited quality baseline.

Both approaches provides improvement on different languages, which indicates the generality of these approaches. For the proposed work, we plan to extend the work on limited resources languages to the open domain, to show that the context knowledge can be beneficial for Speech Retrieval regardless of condition. We also aim to shows that the improvement on the IR system is beneficial to the Speech Retrieval system, and more efforts should be put in towards this direction.

1.4 Summary of Completed Work

We have investigated how to integrate these two approaches to Speech Retrieval, and determined that the approaches can achieve improvement on STD under the limited resources condition. The resulting system has been shown to be effective on multiple languages, implying that the improvement is language independent. The work can be summarized as follows:

- We describe two different sources of context from conversation, Word Burst and Local Knowledge. Both improve STD under the limited resources condition.
- We exploit better search strategy to improve STD. The experiments show that the improvement is neither sensitive to the deployed ASR systems nor languages; it is only sensitive to the query.
- We show that the better search strategy can be integrated with the existing ASR system combination. Our work provides additive benefit with existing ASR system combination.

1.5 Proposed Work

Based on our positive result regarding the limited resources condition, we propose to extend existing approaches for better Speech Retrieval under the open domain condition. We also define a new task, Spoken Snippet Retrieval (SSR), which retrieve a moderate size of speech from the speech collection with just enough context. In this case, user can understand the most essential content for the query among the speech collection, without listening through a long recording.

We also propose to have different evaluation metrics for the completed task. The existing evaluation metrics are designed for a specific task, which is hard to interpret outside of the task. We wish to re-evaluation the result with a more common measurement such as F-score to make the performance more interpretable.

Lastly, we propose to create a theoretical framework for the proposed approach. Right now the improvement are consistent among different languages, yet there is still no way to formula in a more general way. We wish to discover a generalized way to present our work for wider use of our approach.

1.6 Proposal Organization

The reminder of this proposal is organized as follows:

- Chapter 2 introduces the general framework for a Speech Retrieval system. It also explains how ASR and IR system works in Speech Retrieval, which will help us understand standard Speech Retrieval system and our proposal work better.
- Chapter 3 describes related work in the Speech Retrieval field.
- Chapter 4 presents our work on using contexts from conversation for hypothesis rescoring in limited resources condition Speech Retrieval. We perform experiments on different languages under limited resources condition, and using different conversation based knowledge to improve Speech Retrieval performance. We also provide analysis upon the characteristics of query, since the effectiveness of contexts from conversation is sensitive to the query.
- Chapter 5 presents our work on improving IR system for Speech Retrieval in limited resources languages. We look at combination of systems and find the improvement from this approach is also independent with language and ASR systems. It is only sensitive to the query. We further combined this approach with the existing ASR system combination, and the improvement is additive with it.
- Chapter 6 present our proposed work on the open domain condition Speech Retrieval. We plan to extend our positive discovery to the open domain Speech Retrieval with the experiments based on Youtube videos.
- Chapter 7 provide a time line for the proposed work.

Chapter 2

Speech Retrieval Systems

In this chapter, we first present the general framework of a Speech Retrieval System. Then, we separately discuss two major components, ASR system and IR system within this framework. And finally, some important concepts in Speech Retrieval are discussed.

2.1 The general Speech Retrieval framework

Speech Retrieval retrieved segments of audio that is related to the given query from an audio corpus. In the past decades, different algorithms and technique had been studied and developed to improve varies kind of Speech Retrieval tasks, such as Spoken Document Retrieval (SDR) or Spoken Term Detection (STD).

A Speech Retrieval system usually include two major components: the ASR system and the IR system. As shown in figure 2.1, the ASR system decode the speech corpus and generate text based hypotheses to represent the corpus. The IR system perform search on the text representation of speech corpus. It retrieves the appropriate result depending on the given query and the speech retrieval task.

For the task that does not require user query such as Summarization of speech data, the IR system generate the result based on the decoding hypothesis. It still follow this general pipeline of Speech Retrieval system.

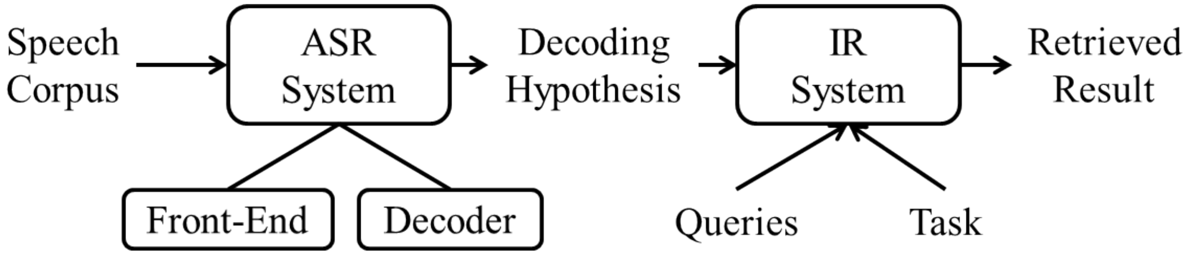

Figure 2.1: Speech Retrieval System pipeline

2.2 ASR system

The ASR system converts spoken words into text. An ASR system genearlly includes two components: The front-end and the decoder. The front-end extracts feature observation from the input speech signal, so as to obtain an appropriate representation of speech. The decoder then output decoding hypotheses according to the feature representation generated by the front-end. The decoding hypotheses can be represented in several format, each format has its own strength and weakness.

2.2.1 Front End

The input speech signal for front-end is usually time-domain sampled speech waveform. This is commonly used for storing speech data, yet it is not ideal for ASR. The ASR system tries to simulate how human hearing worked, which is based on the characteristics of speech sounds in frequency-domain. As a result, a spectral representation of speech signal is more appropriate than the time-domain based representation of speech signal for ASR. Since speech signal is stationary within a short period of time but changes over a longer time [38], we need to segment the input speech signal into small frames when extracting the features. A commonly used frame length is 25 msec, which is considered as short enough to capture the rapid transitions in speech yet has good enough time-domain resolution. The mel-frequency cepstral coefficients (MFCCs) is one of the most popular feature representations in speech recognition, as the mel-scle approximate the human auditory response better. [13]

2.2.2 Decoder

Following the front-end feature extraction, the decoder decodes the most probable word sequence from the extracted feature. There are three major component in a decoder, which are acoustic model, language model and dictionary. For acoustic model, most decoder adopt the hidden Markov models (HMMs) [3, 4] to capture the acoustic characteristics of speech data. The HMM parameters can be estimated by using the Baum-Welch (BW) algorithm [5], a special case of the Expectation-Maximization (EM) algorithm [14]. Language model is for obtaining the prior probability of a specific word sequence in a language. The most commonly used language model is *n-gram* language model. It is very helpful to discriminate acoustic ambiguous speech and reduce the search space while decoding. For example, it is very hard to discriminate the following two utterances, “I OWE YOU TOO” and “EYE O U TWO” from acoustic information. With the language model, we know that the first utterance is more likely to happen in the real life. The dictionary is the third component in a decoder. It is the bridge between acoustic model and language model. While acoustic model and language model works on measuring the different property of speech in a language, dictionary link both models with lexical knowledge. Dictionary provides pronunciations of words, so the decoder know which HMMs to use for a certain word. It also provides a list of words to limit the search space for decoder. As a result, an ASR system can only recognize the word presented in the dictionary. The dictionary for an ASR system usually requires multiple linguists manually write rules and check the pronunciations. Hence, it is very costly and time consuming to create a dictionary for ASR system.

2.2.3 Hypotheses Representation

The ASR system generates recognition hypotheses based on the input speech data, the hypotheses can be represented in different ways. For the Speech Retrieval task, there are three different hypotheses representations that are most common:

- One-best hypothesis
- Lattice
- Confusion Network

One-best hypotheses contain the decoding result with highest probability for the input speech generated by the ASR system. The quality of one-best hypotheses can be estimated by computing the word error rate (WER). It is the most compact representation out of the three, since it only has 1 hypothesis for each time segment. If the WER is low, one-best hypothesis is the best hypotheses representation, since its format significantly reduces the search space for the IR system. One-best hypotheses is widely used in SDR task.

Lattices are representation of the possible recognition word hypotheses. It contains a set of word hypotheses with boundary times and transitions between different hypotheses [35]. It can be found that lattice tends to contain a large number of word hypotheses including both the true hypotheses and the competing hypotheses. If the WER is high for the decoding result, lattice is considered as a better hypotheses representation compare to one-best hypotheses because of the rich information embedded in it.

Confusion networks [29, 30] are another hypotheses representation which is generated from lattices. It is simpler and more compact compare to lattice, since it consist of a number of clusters connected sequentially. Each cluster consists of one or more words associated with probabilities. A word corresponds to one or more arcs in the lattice, and its probability is the sum of the posterior probabilities of the arcs in the lattice. Each cluster has a starting time and an ending time; these are calculated as weighted averages of the starting and ending times of lattice nodes or arcs that correspond to words in the cluster, and then adjusted so that the ending time of each cluster is equal to the starting time of the next cluster.

In general, lattices and confusion networks are considered as better options when the decoding quality is not good, while one-best hypotheses is the best choice when high quality ASR result is available. It is possible to convert the more complicate hypotheses representation into a simpler one, but there are information loss during the conversion.

2.3 IR system

The IR system retrieves result that is relevant to the user's query from the decoding hypotheses. The IR system has different retrieval methodology according to each task. We introduce the IR system for SDR and STD tasks in the following sections.

2.3.1 IR system for SDR task

SDR retrieves spoken documents that is related to the give query from archives of speech recordings. The query in SDR task is natural language *text* queries such as *"What natural disasters* *occurred in the world in 1998 causing at least 10 deaths?"*. One-best hypotheses is the most commonly used hypotheses representation in SDR task, since the previous SDR task is on broadcast news, and the WER is good.

After ASR decoding, preprocessing are applied to clean the recognition hypotheses. For example, case normalization, stemming, stop word removal and sequence word mapping (Mapping sequential words to specific compound word). Text based analysis such as part-of-speech (POS) tagging is applied to the query text to identify the weight of each word in the query text. The retrieval of spoken document is based on the weight for each word in the query, and the distribution of word in the spoken document to rank all the relevant spoken documents. Further document based relevance feedback can be applied to further improve the SDR performance [20].

The key for successful SDR is to capture the keyword in the spoken document. If a spoken document has keyword that's in the query, the spoken document is very likely to be related to the query. As a result, it inspires the later research in STD, which focus on detecting the keyword.

2.3.2 IR system for STD task

STD retrieves the presence of the give query in a speech corpus from archives of speech recordings. The query in STD task is single or multiple text words. The two more complicated hypotheses representation, lattices and confusion networks are commonly used in STD task, since the hypotheses other than the best one is still a valid indicator of the word existence. There are different retrieval applied applied to different hypotheses representation.

If the hypotheses are represented in lattices, a Finite State Transducer (FST) based search is applied to the lattices. The entire retrieval can be separated into two part: Indexing and Search. At the indexing stage, the lattice of each utterance is expanded into a finite-state transducer, such that each successful path in the expanded transducer represents a single word or a sequence of words in the original lattice. The posterior score, start-time and end-time of the corresponding word or word sequence are then encoded as a 3-dimensional weight of the path. At the search stage, in-vocabulary (IV) queries are usually compiled into linear finite-state acceptors (FSA), with zero cost. Out-of-vocabularty (OOV) queries are mapped to IV queries (proxies) [10] according to phonetic similarity, which usually results in non-linear finite-state acceptors with different cost for each proxy. Regardless of being IV or OOV queries, STD is done by composing the query FSA with the index, and one can work out the posterior score, start-time and end-time from the weight of the resulting FST.

The retrieval for confusion networks is carried out in another way. For single-word queries, each occurrence of the query word in the confusion networks generates a detection. The starting and ending times of the detection are those of the cluster containing the word; the score of the detection is the probability of the word. For multiple-word queries, dynamic programming is used to find all paths in the confusion networks such that the words on the path form the query. The paths may contain epsilon words. Each path generates a detection: the starting and ending times are those of the first and last clusters in the path, and the score is the product of the probabilities of all the words (including epsilon words) in the path. If multiple detections for the same query overlap, only the one with the highest score is retained.

The strength and weakness of both approaches are discussed in the fifth chapter. The analysis and combination of these two approaches is one of the main contribution of this proposal.

2.4 Important concepts in Speech Retrieval

Speech Retrieval is an application of ASR technique. However, there are several important concepts in Speech Retrieval which might not be identical as in ASR. That is due to the difference between Speech Retrieval and ASR. The focus for these two tasks are different.

2.4.1 WER is no longer the holy grail for Speech Retrieval

WER is the main evaluation metrics used in the ASR task. It measures how many errors are made by the ASR system. It is calculated by computing the word based Levenshtein distance between the 1-best hypothesis from ASR system and the reference transcript.

In order to evaluate how good the ASR system performs in Speech Retrieval, WER is usually positively correlated with the retrieval performance, yet it is not the only valid evaluation metrics. If the hypotheses representation is using lattices or confusion networks, which does not only contain the the one-best hypotheses, the WER can only show the quality of a small portion of the hypotheses. The metric that can evaluate the quality of the entire hypothesis representation such as lattices/confusion networks recall can provide better insight on the ASR output for Speech Retrieval, since the retrieved result can come from every part in the lattice/confusion networks.

2.4.2 OOV words can still be retrieved

OOV word is a critical issue in ASR. ASR systems can not recognize word that is not in their dictionary. However, it is possible to retrieve result from OOV query, since the query is already given. We can change our retrieval strategy based on the query type. If the query is OOV, we can use phonetic/morpheme based search or proxies based search which does not require exact query word exist in the ASR hypotheses.

2.5 Summary

In this chapter, we discussed the general framework and two major components of a Speech Retrieval System including ASR system and IR system. This prepares the required knowledge for understanding the proposed work.

Chapter 3

Related Work in Speech Retrieval

In this chapter, we discuss the related work for the topic discussed in this proposal, including the current solutions for different tasks in Speech Retrieval and the previous work in using context for language processing. We first go through the progress in SDR tasks, which later inspire the STD task. Then we go through the progress for STD task, from the rich-resources language to the limited resources languages. Lastly, we review work in applying context for language processing, since it is correlated to our approach for the STD task.

3.1 Progress in the SDR task

SDR task was part of the TREC evaluation around 2000. As a result, different groups around the world participate in the evaluation and provide useful methods towards effective SDR performance. [40] analysis the role of term frequency (TF) in SDR tasks. It attempted tf*idf a more theoretic interpretation and point out a possible reason why multiplying tf directly with idf causes the poor performance with the given interpretation. [20] create a system based on Okapi scheme based retrieval, stemming, part of speech weighting of query term, stemmer exception list, parallel collection frequency weighting, relevance feedback and document expansion. [42] uses a hybrid IR-linguistic system, following by a very effective document expansion technique [41, 43] for the SDR task. [18] also reported their SDR system, they conclude the SDR system based on ASR transcript are quite close the the average precisions obtained on manual transcripts, indicating that the transcription quality is not the limiting factor on IR performance.

Most of these works are more focus on the IR perspective of Speech Retrieval task. IR technique such as query expansion, document expansion are commonly seen in these works. On the other hand, ASR systems are not the main focus in these work. Most systems just treat the one-best hypotheses from the ASR system as a “noisy text”, and perform regular IR technique based on those.

An important knowledge the Speech Retrieval society learned from the SDR tasks are: If the speech segment that contains the exact keyword/query can be correctly recognized and retrieved, just retrieve that segment. It usually has very good precision regarding the query term, and the recall is harder to evaluate in real world retrieval task (Since there is no reference for every speech segment in the corpus). Just find the keyword and retrieve it can achieve a very decent result in

SDR. This inspire the Speech Retrieval community into the next stage: STD, which focus on detecting the presence of the exact query word. This also shift the focus for the Speech Retrieval from IR component to ASR, since detecting the existence of word is heavily rely on the ASR system to correctly recognize the keyword.

3.2 Progress in the STD task

STD task was define at around 2006 [16]. It focus on detecting the existence of word instead of retrieving the spoken document. Since the ASR performance is very sensitive to the quantity of speech data available for training, the progress in STD task can be separated into two stage: rich resources condition and limited resources condition. The rich resources STD is carried out under English, Chinese and Arabic, which are considered as the language with more resources available. The STD system under this condition does not have limitation on the amount of training data used for training ASR systems. As a result, a higher quality recognition result can be expected in rich resources condition. The success in rich resources condition later lead to the STD under limited resources conditions. In the limited resources condition, the training data for ASR system is limited to 10 hours, which leads to a very high WER. It is also being tested on the language that does not have huge amount of linguistic resources available such as Tagalog, Cantonese, Assamese, etc. The bad recognition hypotheses heavily affect the performance from the IR system. As as result, research efforts are done to recover the damage from the limited resources condition to make it as good as the rich resources condition.

3.2.1 Rich resources condition

The STD work in rich resources condition establish the standard pipeline for STD systems, since the ASR and IR two stage approach has overall better performance compare to other approach such as directly search on the acoustics. [34] present a STD system which search on word lattices. It estimates word posteriors from the lattices and uses them to compute a detection threshold that minimizes the expected value of a user-specified cost function. For the OOV query, the system uses approximate string matching on induced phonetic transcripts. [26] present a vocabulary independent system that can process arbitrary queries, exploiting the information provided by having both word transcripts and phonetic transcripts. This system is based on word confusion networks and phonetic lattices. [46] reported their system for the STD 2006 task, they also analysis the effectiveness of different index ranking schemes, and the utility of approaches to deal with OOV terms.

Despite these systems are different in detail, the structure are similar. The ASR system and IR system are two indispensable component for a successful STD systems. Lattices and confusion networks are start being used as the ideal hypotheses representation instead of the one-best hypotheses, due to the rich information contains in them. Different search strategies are applied to the IV and OOV queries respectively to compensate the inability of ASR system for recognizing OOV words. These discovery are still valid today, every state of the art STD system uses the similar pipeline and strategy for STD task.

The STD performance in rich resources is very good. The best system can achieve around 80% of the maximum possible accuracy score in English. However, the good result is based on high quality ASR output. What if the high quality ASR result is not available? This question leads the STD research into the next stage: limited resources condition.

3.2.2 Limited resources condition

STD under limited resources condition is proposed as a research challenge recently. It is one of the focus in current speech community. The ASR system with limited training data is unable to generate high quality decoding result. Hence, the low quality hypothesis representation limit the achievable performance for the IR system, and ended up having far worse STD performance compare to STD under rich resources condition. There are multiple works being proposed to improve STD under limited resources condition regarding ASR and IR systems.

Deep Neural Network (DNN) is widely used in ASR recently. [12] first propose a novel context-dependent (CD) model for ASR. It introduce a pre-trained deep neural network hidden Markov model (DNN-HMM) hybrid architecture that trains the DNN to produce a distribution over senones (tied triphone states) as its output. It provides significant improvement on the regular ASR task. [32, 33, 49] further extend the usage of DNN with dropout and maxout technique to make it further robust under limited resources condition. Another approaches is to include the multilingual information, using the DNN from a rich resources language to improve limited resources language's performance [22]. All of these works provide solid improvement for ASR under limited resources condition, yet the WER is still very high, due to the limited resources condition severely degrade the decoding quality.

For the IR systems, there are two lines of research that dedicate to better STD performance under limited resources condition. The first line focuses on rescoring the recognition hypotheses according to different feature or information to obtain better hypotheses representation. [28] used transformation-based learning and lexical features to improve WER from the two best hypothesis in a CN confusion bin. Similarly, [1] detects errors on broadcast news transcriptions using lexical, syntactic and contextual information. [45] trained conditional random fields using CNs instead of the 1-best transcription to improve accuracy in slot-filling in semantic frames. [44] used syntactic and prosodic features to identify mis-recognized words to generate clarification questions in speech-to-speech translation. [11] proposed graph-based re-ranking for STD by using acoustic similarity. [27] proposed hypotheses rescoring algorithm based on the other retrieval result from the same query. The concept for most of these works are the same: Since the quality of hypotheses is bad, applying other knowledge or information source are beneficial for fixing the error created by the bad ASR system. The other line of research works on combining the result from multiple systems. The diversity from different systems can be integrate together to create a better result compare to each of the individual systems. [31] produce complementary STD systems and show that the performance of the combined system is 3 times better than the best individual component. [27] investigate the problem of extending data fusion methodologies from Information Retrieval for Spoken Term Detection on limited resources condition. [21] perform system combination , where the detections of multiple systems are merged together, and their scores are interpolated with weights which are optimized using the evaluation metrics. These works shows the integration of diverse systems can contribute to a better overall performance in

STD.

The proposed work follows the spirit these two lines of research in STD under limited resources condition. We apply contexts from conversation for hypothesis rescoring, and we improve IR system by search combination, which is system combination based on same decoding result. We aim to extend these works to open domain conditions.

3.3 Applying context in language processing

There are earlier works in applying context for language processing. [24] introduces two complementary models that represent dependencies between words in local and non-local contexts. The type of local dependencies considered are sequences of part of speech categories for words. The non-local context of word dependency considered here is that of word recurrence, which is typical in a text. [23] present a language model which reflects short term patterns of word use by means of a cache component, combining with the traditional trigram language model. [19] describe a simple model based on the trigram frequencies estimated from the partially dictated document which cache the recent history of words. [39] describe a model that attempts to capture within-document word sequence correlations.

These works focus on modeling the sequence of word token in different way. The modeling requires correct transcript or text to calculate probability, and most of the work is done on text corpus or news speech corpus such as Wall Street Journal, which are all well-structured (spoken) documents. Our work focuses on the time difference instead of the token sequence. It is based on the context from conversation, which is multi-speaker and less-structued speech. Our approach does not require any correct transcriptions to generate context based models, instead, it focuses on applying the context from the noisy ASR decoding hypotheses. The detailed difference will be discussed in the section where we introduce our approach.

3.4 Summary

In above paragraphs, we discussed the current solutions for different tasks in Speech Retrieval and previous work in using context for language processing. To achieve even better performance, we believe the context from our understanding of conversations should be integrated, instead of only using the word tokens for retrieval. Also, most of the system combination nowadays focusing on combining different ASR system output. We believe IR system also plays an important role, and combining from the IR system can provide further improvement, which is complementary with the existing ASR system combination.

Chapter 4

Contexts from Conversation for Hypotheses Rescoring

In this chapter, we study how to apply contexts from conversation for hypotheses rescoring under limited resources condition. We explore two different approaches for hypotheses rescoring, which are Local Knowledge and Word Burst. We first present the definition of each contexts, then we compare our approaches to the traditional context based approach for language processing. We design hypotheses rescoring algorithm based on different context we presented. Following that, we report experiments on multiple dataset with the rescoring algorithm under limited resources condition. After the experimental results, we provide detailed analysis on the result we obtained. In the end of chapter, we propose to evaluate in different metrics and develop a more theoretical framework for these approaches.

4.1 Hypotheses rescoring with Local Knowledge

4.1.1 Local Knowledge

The assumption of Local Knowledge is as follows : Any word that actually occurs in a local context is expected to be recognized confidently at least once. We examine three different levels of local context: corpus, conversation and speaker. We observe that conversation is the best level of local context for STD. Based on Local Knowledge, we propose an algorithm called Local Rescoring that improves STD under limited resources condition.

How Local Knowledge different from the previous context based work

Local Knowledge shares the concept of previous work that the same word is likely to reoccur in the same “document”, yet we varies the size of “document” in three different level to examine which is the most ideal setup for conversation speech. Also, Local Knowledge relies the context in conversational setup, which has multiple speakers and less structured speech. The context based research before focus on text documents or well-structured news speech, which only has single speaker. Last but not least, traditional work on applying context requires a text corpus or transcribed speech data (Which has no errors) other than the decoding target to compute the probability of contexts, while our work is focusing on the context of noisy, recognized and not completely correct hypotheses. This distinguish our work from the traditional context based approach, since we believe applying the context from our recognized result has less domain mismatch compare to modeling the context from other source.

Three levels of Local Knowledge

Local Knowledge asserts: If a keyword is really present, it is likely to have at least one instance with high confidence score at a given level of local context. Otherwise, the instances of the keyword are all recognized with low confidence score only, which is likely the result of recognition errors. We define three different levels of local context:

- *Corpus*: If a keyword is recognized confidently in the entire corpus once, we classify the low confidence score instances of the keyword in the corpus as correct recognition results.
- *Conversation*: If a keyword is recognized confidently in a conversation once, we classify the low confidence score instances of the keyword in the same conversation as correct.
- *Speaker*: If a keyword spoken by a speaker is recognized confidently once, we classify the low confidence score instances of the keyword spoken by the same speaker as correct.

4.1.2 Local Rescoring

Local Rescoring uses the Local Knowledge we described in the previous section for hypothesis rescoring. According to Local Knowledge, we can partition all recognition hypotheses into two groups: instances of keywords being classified as recognized correctly, and those classified as recognition errors. For a word classified as a recognition error, we apply a penalty to its recognition confidence score.

Local Knowledge relies on the word with high confidence word to prevent the same word from being classified as a recognition error. The high confidence word is decided by a sentence-based threshold. The threshold is the average of top 50% highest posterior probabilities for hypothesis per utterance. Hence, the threshold changes according to different utterances. This dynamic threshold can preserve high confidence word from every utterance, instead of only favoring some well recognized utterances. The words are classified as recognized correctly if they have an instance above the threshold within the same level of local context. By processing all hypotheses, every word is classified as either recognized correctly or as a recognition error. The rescoring follows a simple formula:

For each word that was classified as potentially a recognition error:

$$p'(w) = p(w) * penalty(L)$$

In the formula, $p'(w)$ is the new confidence score after Local Rescoring. $p(w)$ is the original confidence of score before Local Rescoring. $penalty(L)$ is the language specific penalty that we obtained from tuning on development data. The confidence score for the word being classified as a correct recognition result does not have to be changed during the Local Rescoring.

4.2 Hypotheses rescoring with Word Burst

4.2.1 Word Burst

We observe that conversations tend to focus on particular topics, the high likelihood of a word related to the current topic occurring near other instances of the same word is called Word Burst. More precisely, when in a conversation that touches on specific topics, the content word within the same topic will tend to occur near each other. We take an existing vocabulary and (limited) text resources and use it to define a stop list as the most frequent words in the available corpus; we experimented with lists that include 1 - 5% of the vocabulary, and the word not in the stop list is considered as content word. We design a rescoring algorithm based on Word Burst, and further present a target expansion procedure for Word Burst rescoring that extends the hypothesis used in rescoring to variants of the same word.

How Word Burst different from the previous context based work

Word Burst captures the relationship between same word during conversation. It follows the similar intuition with the previous work on exploiting context for language modeling, yet there are a few major difference. First, most of the work done before relies on a good amount of text/transcription for modeling the context between words, while Word Burst only requires limited amount of text for deciding stop word list. Since the goal for stop word list is to filter away the most frequent word in the target language, even with a limited amount of data, the word frequency follows the same trend. Second, Word Burst utilize the context of temporary topic during conversations, which is depend on the time distance instead of the token distance. This capture the degradation of specific topic regarding times, since the silence during conversation leads to the end of a topic. While the traditional work focuses on the token distance, it does not put any focus into time and silence. A 20 second silence is ignored in the previous works, yet it means the end of current topic in the Word Burst set up. Last, the source of context are different between Word Burst and the traditional context works. The traditional works obtain context from a good amount of text document or well-structured spoken document, where the source of context is good at quality and quantity. Word Burst employ the context from the noisy recognition hypotheses, and only the word within few second of each utterance. The quality and quantity of available context is bad, yet it can still contribute to impressive improvement in Spoken Term Detection.

Evidence of Word Burst

In this section, we showed several evidence in our data which suggest Word Burst exist in the conversational speech.

[2] proposed using a window size of 20 sec to detect the topic state in meetings; we used this as a starting point for identifying Word Burst. Table 4.1 shows the percentage of content word within burst windows. We exclude words from the top 1% and all singletons in the available corpus. As can be observed, content words (as defined) tend to occur in bursts.

Figure 4.1 provides a visual example, using the distribution of Tagalog term *magkano* in

Table 4.1: Content word window size / burst percentage.

	10sec	15sec	20sec	25sec	30sec
Cantonese	43.6	48.4	51.3	53.2	55.0
Pashto	35.7	40.2	43.3	45.7	47.9
Tagalog	40.7	45.0	48.0	50.0	51.6
Turkish	35.4	39.2	41.4	43.1	44.4

our data. For the graph, we can see the word have the tendency to occur in bursts. This needs to vary according to language. In an agglutinative language such as Turkish, there are many morphological variants and this required a longer stop-word list.

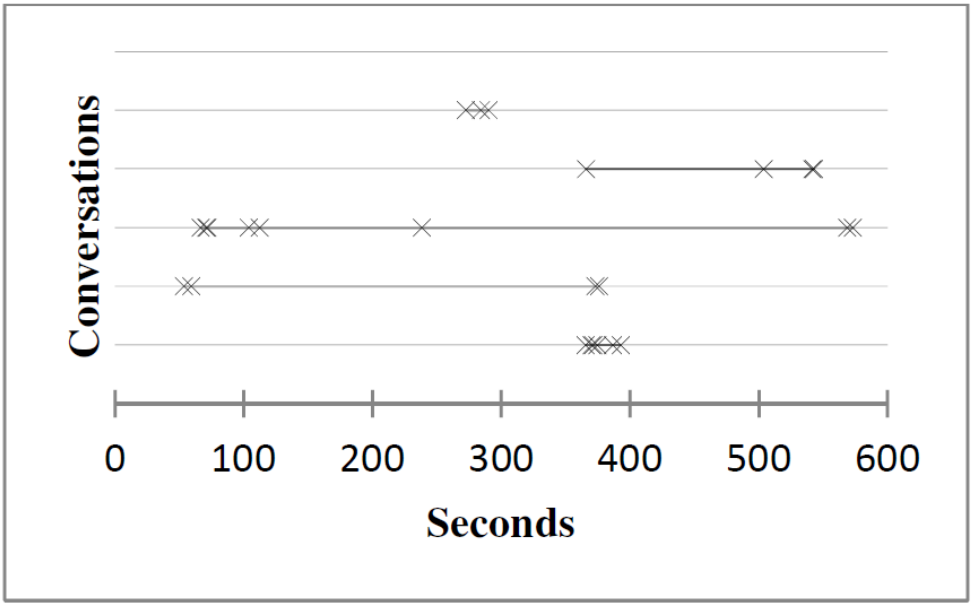

Figure 4.1: Term incidence for Tagalog *magkano* (abscissa is time in sec; individual lines are separate conversations)

4.2.2 Word Burst Rescoring

The general concept of Word Burst rescoring includes two parts. For each hypothesis, if there is another hypothesis of the same word temporally close, it gets a score bonus; if there is no other hypothesis of the same word close to it, it receives a penalty. There are exceptions for each case, which is described in the following graph.

Figure 4.2 illustrates Word Burst rescoring. The y-axis in this figure shows the probability of each hypothesis. There are three hypotheses A, B and C to which we apply our rescoring algorithm. There are hypotheses A1 and A2 which are the same word as A. Both of them occur near hypothesis A in time. Hypotheses B and C have no nearby hypotheses.

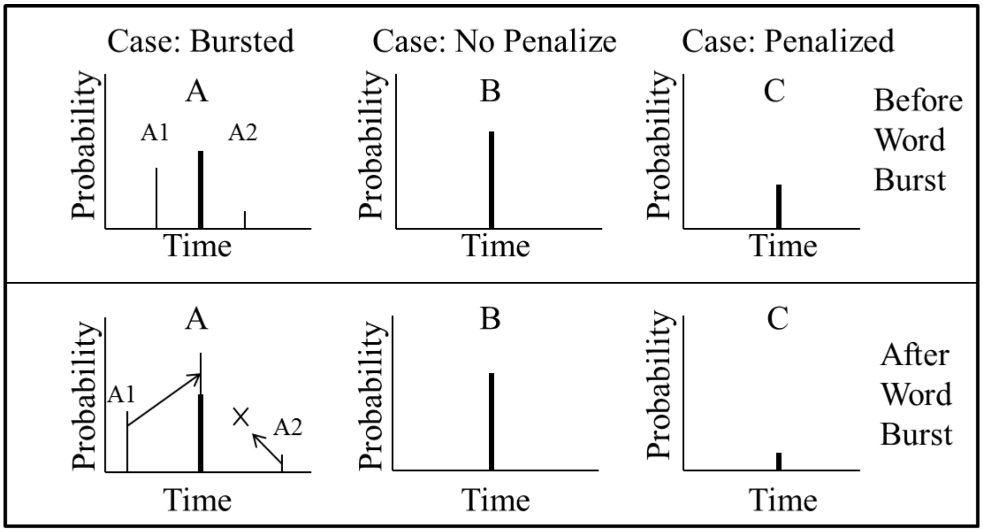

Figure 4.2: Concept of Word Burst Rescoring

The transition in the figure shows how the rescoring algorithm works. There are three possible cases which are represented by hypothesis A, B and C for Word Burst Rescoring. For hypothesis A, the probability of A is boosted in proportion of A1s probability. Although hypothesis A2 also occurs near A, the probability is below a threshold, possibly because A2 might be a recognition error. Hence, hypothesis A2 does not boost A. Both hypothesis B and C are isolated, with no same-word neighbors. Since hypothesis B already has high probability, we assume that it is a correct recognition. However, for hypothesis C, we penalize the probability, since it violates the Word Burst assumption.

The algorithm use probability from hypotheses to decide whether to apply rescoring or not. This modification addresses some issues that might happen in the Word Burst assumption:

- Some hypotheses in the decoding result are recognition errors. Boosting probability according to recognition errors degrades the quality of decoding result.
- Some words do occur alone. These words can get very high probability during decoding. Penalizing all isolated word harms performance.

The following formulas show the algorithm:

For each word x, x_i and x_j are two different hypotheses of the same word. The $p'(x_i)$ is the probability after rescoring, and the $p(x_i)$ is the probability before rescoring. If there is no x_j that occurs close to x_i, and the probability of x_i is below the penalty threshold pt, then $p'(x_i)$ can be computed as:

$$p'(x_i) = p(x_i) * penalty(L)$$

where $penalty(L)$ is the language-dependent penalty for each non-burst word.

If there is x_j that occurs near x_i, and the probability of x_j is above the bonus threshold bt, then $p'(x_i)$ can be computed as:

$$p'(x_i) = p(x_i) + b(x_i)$$

$b(x_i)$ is a bonus function computed from x_i

$$b(x_i) = (\sum_j w(x_i, x_j) * p(x_j)) * e^S$$

$w(x_i, x_j)$ is the weight for Word Burst between x_i and x_j, S is the sum of all weight for differnet x_j.

$$w(x_i, x_j) = 1 - (dis(x_i, x_j) - windowsize)$$

$$S = \sum_j w(x_i, x_j)$$

$dis(x_i, x_j)$ is the time distance between instance x_i and x_j. $windowsize$ is the maximum time interval for a Word Burst. The thresholds pt, bt and $windowsize$ are computed from the development set.

4.2.3 Target Extension for Word Burst

Word Burst uses reoccurrence of a word hypothesis for rescoring but relies on the reoccurrence of the exact word, a problem in agglutinative languages where the hypothesis may reoccur, but as a morphological variant. This phenomenon limits the detection of Word Bursts.

One solution is to find a way to extend the target set used for Word Burst rescoring. The hypothesis can then be rescored based on the occurrence of hypotheses that belong to the set. There are several ways to create hypothesis sets; we want to focus on simple language-independent approaches, fitting the limited resources theme.

Substring-based target extension

A simple approach, for language with alphabetical writing systems, is to use sub-string overlap. Each hypothesis is grouped with the hypotheses that share a substring. This method accounts for some morphological variation, since the hypotheses will likely share characters. For example, the hypothesis *prepared* and *preparation* share a substring of *prepar*. The substring technique does not require language-specific knowledge, which makes it easier to apply to alphabetic languages. For agglutinative languages we can tune substring length on development data.

Morphology-based target extension

As a comparison we evaluate the use of language-dependent morphology to perform target extension. That is, we segment each word in our dictionary into multiple sub-word segments. For example, the word *unfriendly* is segmented into *un-friend-ly*. We form sets for Word Burst target extension according to the overlap of sub-word segments.

4.3 Dataset and experiments setup

4.3.1 Dataset

We use speech from five different languages: Cantonese, Pashto, Tagalog, Turkish and Vietnamese, as provided by the IARPA BABEL program. As for Word Burst target extension experiment, we also included Zulu data since we wish to confirm the effect on multiple agglutiative language. Each language has 10 hours of training data and 10 hours of development data. We use 5-fold cross validation (8 hours of development and 2 hours of testing data) for parameter tuning and evaluation.

4.3.2 Experiments setup

STD system description

Our STD system uses a ASR-IR two stage pipeline. The decoded hypotheses are represented as confusion networks. Confusion networks are generated from the combination of three different decoding systems [15]. Both Local Rescoring and Word Burst Rescoring are applied to every hypothesis in the confusion network. Our IR component output the location of queries in the confusion network, yet it skips OOV queries and only retrieve result for IV queries.

Evaluation Metrics

We use Actual Term Weighted Value (ATWV) [16, 47] for evaluation. ATWV is the average of TWV over all queries. The formula for TWV is as follows:

$$TWV(\theta) = 1 - (P_{\text{Miss}}(term, \theta) + \beta * P_{\text{FA}}(term, \theta))$$

where θ is the detection threshold, β is a factor that controls the balance between misses and false alarms, set to 999.9.

The concept of TWV score is simple: If the system performs perfectly on a query, it has a TWV of 1; if the system misses some of the query words or produces false alarms, it receives penalty on the TWV score. As a result, the TWV score is bounded above by 1 but has no fixed lower bound.

Original query set and Non-singleton query set

ATWV is very sensitive to the characteristics of queries in the query set. For a query with only 1 occurrence in the testing data, the TWV gain from a single correct detection is equal to the penalty received from generating 36 false alarms. For queries with multiple instances, this correct detection/false alarm differences are closer. The query set used in different languages affects the performance of Local Rescoring. Both Local Rescoring and Word Burst Rescoring do not support queries that only occur once in the entire testing data, so-called singleton queries. However, the percentage of singleton queries in the original query set differs in each language. These are listed in Table 4.2. A 45% singleton query rate means that 45% of the queries occur only once in testing data. Accordingly, we also remove all singleton queries and present non-singleton query set results separately.

Table 4.2: Singleton Query Distribution in different test languages

Language	Singleton Query %
Cantonese	45
Pashto	35
Tagalog	38
Turkish	50
Vietnamese	54

4.4 Experimental results

4.4.1 Local Rescoring Result

The Local Rescoring experiments are conducted on five languages. For each language, we perform the experiment with all three levels of local context (Corpus, Conversation and Speaker). We also report and compare the results on two different query sets: the original query set and the non-singleton query set. We expect the non-singleton query set to provide better insight into Local Rescoring, since the difference on query set are eliminated.

Table 4.3: ATWV for different levels of local context with original query set

Language	Baseline	Speaker	Conversation	Corpus
Cantonese	0.114	0.116	0.116	0.113
Pashto	0.073	0.093	0.094	0.073
Tagalog	0.136	0.163	0.161	0.140
Turkish	0.241	0.242	0.242	0.241
Vietnamese	0.085	0.081	0.085	0.083
Mean	0.130	0.139	0.140	0.130

Table 4.4: ATWV for different levels of local context with non-singleton query set

Language	Baseline	Speaker	Conversation	Corpus
Cantonese	0.107	0.114	0.117	0.106
Pashto	0.067	0.093	0.094	0.067
Tagalog	0.134	0.167	0.166	0.137
Turkish	0.227	0.228	0.229	0.227
Vietnamese	0.079	0.081	0.084	0.077
Mean	0.123	0.137	0.138	0.123

Tables 4.3 and 4.4 compare three different levels of local context in five different languages, with two different query sets. Local Rescoring produces limited improvement on Turkish, because it is an agglutinative language. Local Knowledge does not affect morphological variant of words. The Corpus setup shows the least improvement. We discuss this in the Analysis part. The Speaker and Conversation are more appropriate levels of context. Both levels of local context provide similar improvement in ATWV, although Conversation provides more consistent improvements. We conclude that Conversation is the most appropriate level of local context for the STD task.

Table 4.5: ATWV relative improvement (%) on different query sets in Conversation setup

Language	Original	Non-singleton	Singleton Query %
Cantonese	+1.8	+9.3	45
Pashto	+28.8	+40.3	35
Tagalog	+18.4	+23.9	38
Turkish	+0.4	+0.9	50
Vietnamese	+0.0	+6.3	54
Mean	+7.6	+12.1	44.4

In Table 4.5, we show the relative improvement in ATWV on the Conversation level using different query sets. The improvement on non-singleton query set is more consistent and higher. Cantonese and Vietnamese have the most distinctive difference between original query set and non-singleton query set. This difference is due to the high singleton query percentage in the original query set. By eliminating the difference on the query set, we can observe unbiased performance improvement on ATWV with Local Rescoring.

4.4.2 Word Burst Rescoring Result

For the Word Burst Rescoring, we also provide experiments on the same five languages with two different query sets.

Table 4.6: ATWV for Word Burst Rescoring with original query set

Language	Baseline	Word Burst	Δ ATWV (%)
Cantonese	0.114	0.122	+7
Pashto	0.073	0.103	+41
Tagalog	0.136	0.173	+27
Turkish	0.241	0.245	+2
Vietnamese	0.085	0.088	+3
Mean	0.130	0.146	+12

Comparing Table 4.6 with Table 4.7, we can observe the improvements in non-singleton queries are larger and more stable. This indicates Word Burst is more useful in the condition without singleton queries. The Vietnamese has the largest ATWV difference between original

Table 4.7: ATWV for Word Burst Rescoring with non-singleton query set

Language	Baseline	Word Burst	Δ ATWV (%)
Cantonese	0.107	0.118	+10
Pashto	0.067	0.101	+51
Tagalog	0.134	0.180	+34
Turkish	0.227	0.230	+1
Vietnamese	0.079	0.088	+11
Mean	0.123	0.143	+16

query set and non-singleton query set, since it has the highest singleton queries percentage. The Turkish performance is still limited because it is agglutinative language.

Word Burst Rescoring generally shows better improvement compare to Local Rescoring, the major difference between two algorithm will be discussed in the analysis section. Also, both Local Rescoring and Word Burst Rescoring have very limited improvement on Turkish. We anticipate this is an issue on agglutinative languages, which needs an solution.

4.4.3 Word Burst Target Extension Result

We tried target extension on two different agglutinative languages, Turkish and Zulu. For each language, we show the result with two different target extension approaches: Substring-based target extension and Morphology-based target extension. This time we only do on the original query set since we expect the target extension can overcome singleton queries by extention from the non-singleton queries.

Table 4.8: ATWV comparison between target extension approaches

Language	Baseline	SubString	Δ ATWV (%)	Morphology	Δ ATWV (%)
Turkish	0.241	0.252	+5	0.245	+2
Zulu	0.115	0.122	+6	0.120	+5
Mean	0.123	0.187	+5	0.183	+3

Table 4.8 shows how different target extension approaches affect ATWV. The proposed substring-based extension outperforms the morphology-based target extension, although both provide improvements on ATWV. Thus target extension can restore the Word Burst effect. Interestingly, a simple procedure that uses an orthographic substring match works better than the morphological decompositions we used. There are more analysis for target extension in the Analysis section.

Table 4.9: Tradeoffs between Correct Detections and False Alarms in Local Rescoring and Word Burst Rescoring (Change in %)

Language	Local		Word Burst	
	CD	FA	CD	FA
Cantonese	-7.9	-25.7	-7	-28
Pashto	-9.1	-33.4	-3	-33
Tagalog	-10.4	-42.0	+0	-35
Turkish	-0.5	-3.1	+2	+4
Vietnamese	-1.7	-11.4	+8	-4
Mean	-5.9	-23.1	+0	-19

4.5 Analysis

4.5.1 Tradeoffs between Correct Detections and False Alarms

Table 4.9 shows that both Local Rescoring and Word Burst Rescoring contribute to significant amount of false alarm reduction. The major difference is Local Rescoring tend to "overkill" correct detections while Word Burst rescoring has better mechanism to preserve the correct detection. The threshold for deciding whether to apply Word Burst rescoring avoid removing correct detection, yet sacrificed some of the false alarm reduction power. However, in our evaluation setup, it values finding correct detection much more than reducing false alarm. This reflects on the ATWV difference, that Word Burst rescoring generally has better ATWV improvement compare to Local Rescoring. Although Word Burst rescoring reduced less false alarm, the correct detection it preserve contribute to higher ATWV score. The source of improvement for both approach are similar: The false alarm reduction.

The only exception in Table 4.9 is Turksih, which has different patterns than every other languages. The identical word based processing does not work as well agglutinative language. Hence we performed the target extension, and the correct detection / false alarm trade off is shown at the following table.

Table 4.10: Correct Detection/ False Alarm tradeoff for two target extension approaches (Change in %)

Language	Substring		Morphology	
	CD	FA	CD	FA
Turkish	+3	-5	+1	-5
Zulu	+1	-18	+1	-16
Mean	+1	-18	+1	-11

Word Burst with target extension also contributes to additional correct detections and reduced false alarms, as shown in table 4.10. This indicates that target extension can recover the utility of conversation structure knowledge in agglutinative languages, since the pattern in agglutinative languages with target extension is the same as the pattern with regular Word Burst in non-agglutinative language.

4.5.2 Words classified as recognition errors in Local Rescoring

Table 4.11: Percentage of words being classified as recognition errors at different levels of local context

Language	Speaker	Conversation	Corpus
Cantonese	35.2	28.3	5.7
Pashto	41.4	34.8	8.2
Tagalog	50.8	42.8	9.7
Turkish	64.3	57.5	21.0
Vietnamese	43.4	36.1	8.4

Table 4.11 shows the percentage of words being classified as recognition errors in different setups. These words are the main focus for Local Rescoring. Except Turkish, all languages have similar trends. Turkish is out of this pattern due to its morphological variation. The variation reduces the classification performance, since the classification process does not consider morphological variants for words being recognized confidently. This leads to a high percentage of words being recognized as recognition errors in Turkish. The Corpus level only has a small portion of words being classified as recognition errors, and the improvement on ATWV is also very limited. This leads to an observation: Defining a large local context that includes too many confidently recognized words weakens Local Rescoring. The Corpus level of context classifies most of the words in the corpus as being recognized correctly. Consequently, Local Rescoring only works on a small portion of data, and does not sufficiently impact ATWV.

4.5.3 Unsuccessful Word Burst target extension

We investigated other possible approaches to target extension. Since Word Burst relies on context, we examined other context information.

One source of context is Mutual Information (MI). We compute the pairwise MI for every word that occurs in the training corpus within the selected window size. The window size is the same as previously used in Word Burst rescoring. We include this MI information in the rescoring process. The words that co-occur in the training data receive bonus, while the words that never co-occur in the training data are penalized. This did not work, as the MI we computed is based on very sparse data, computed from a limited training corpus (10 hours). This approach hurts correct detections, since the co-occurrence distribution is different from training data.

We also examined Brown clustering [7] on the training corpus. Brown clustering places all words in the training corpus into several clusters. We extend the Word Burst target to other words in the same cluster. This does not improve ATWV, and is likely due to limited training data.

Finally, we tried to integrate LDA-based [6] topic modeling for target extension. We computed topic distributions and picked the top X words from each topic. Among these selected words, each word has target extension to other selected words in the same topic. Inspection of the topic word sets does not show a useful relationship, and the ATWV does not increase.

4.5.4 Substring-based Target Extension for Word Burst on Other Languages

We also conducted experiments for substring-based target extension for Word Burst on non-agglutinative languages. The experiments show that target extension on non-agglutinative languages does not outperform regular Word Burst rescoring. If we set the length of substring too short, target extension triggers too many words, leading to false alarms. On the other hand, if we set the length of substring too long, very few substring matches are possible and it has performance similar to Word Burst rescoring, since the target extension does not happen often enough with a requirement of a long substring. As a result, we conclude that while target extension is beneficial for agglutinative languages, for non-agglutinative languages we can simply apply Word Burst rescoring.

The morphology of Turkish and Zulu are both prefix and suffix based. This matches our assumption in proposing substring-based target extension. For languages that have infix based morphology, we expect the improvement from substring-based target extension to be limited.

4.6 Proposed Work

4.6.1 Evaluation with F-score

ATWV is commonly used in the current STD research. However, ATWV relies on a fixed parameters to decide the weight between correct detections and false alarms. It is also sensitive to the quality of query, which makes it even harder to interpret the result. As a result, it is difficult to compare the ATWV values with other work in retrieval field.

We propose to report the current result with F-score, which is a common metric in the retrieval domain. We can calcuate the F-score for each query in the STD result, then average the F-score among every queries. In this case, we can obtain a F-score result which is easy to understand and compare to other existing works.

4.6.2 Develop generalized framework for context from conversations

After our positive discovery in the STD task under limited resources condition, we want to develop a generalized framework for the propose approaches and algorithms. We believe context from conversation should be beneficial to the task that is related to conversations. Hence, we propose to construct a theoretical framework for using context from conversation to describe our completed experiments. The goal for it is to have a mathematical descriptions of our approach to make it more generalized and easier to understand.

4.7 Summary

In this chapter, we studied how to apply contexts from conversation for hypotheses rescoring under limited resources condition. We explore two different approaches for hypotheses rescoring, which are Local Knowledge and Word Burst. Both approaches contributes to better STD performance under limited resources condition, yet Word Burst are considered better since it can effectively preserve the correct detection. We also develope a target extension technique that can apply to Word Burst rescoring, which extend the effect to agglutinative languages. The presented work conclude that using context from conversations can be beneficial to STD under limited resources condition. For the next step, we propose to evaluate in different metrics and create a more theoretical framework for these approaches.

Chapter 5

Analysis and Improvement for STD IR systems

In this chapter, we present analysis and improvement for the current STD IR systems. We first present the two different IR system in our STD system pipeline, which are based on FST search and CN search respectively. After the description of two different search, we present an improved IR search strategy, which is based on the IR system combination, and the motivation for IR system combination. Why it is a special topic we should investigate, and what is it different from the existing system combinations for STD? After the motivation, we demonstrate how the results from different IR systems are combined, following by experiments on multiple dataset under limited resources condition. After the experimental results, we provide detailed analysis on the result we obtained. In the end of chapter, we propose several works to be done in the future. We propose to develop better evaluation metric and theoretical framework for better understanding of the improvement we achieved. Also, we will do more analysis on "*how*" the improvement are gained from the combination, in order to figure out the essential element for a good IR system for Speech Retrieval task.

5.1 Search Description

5.1.1 FST Search

Our FST search pipeline is described in [9, 10], which is capable of both IV and OOV search. We implement the lattice indexing algorithm proposed in [8] making use of the Kaldi toolkit [36].

At the indexing stage, the lattice of each utterance is expanded into a finite-state transducer (FST), such that each successful path in the expanded transducer represents a single word or a sequence of words in the original lattice. The posterior score, start-time and end-time of the corresponding word or word sequence are then encoded as a 3-dimensional weight of the path. Our implementation of the indexing algorithm relies on the fact that the lattices are defeminized at the word level, which is an essential part of our lattice generation procedure [37]. Otherwise the indexing algorithm tends to blow up since the number of potential word sequences grows exponentially with the sequence length.

At the search stage, IV keywords are usually compiled into linear finite-state acceptors (FSA), with zero cost. OOV queries are mapped to IV queries (proxies) [10] according to phonetic similarity, which usually results in non-linear finite-state acceptors with different cost for each proxy. Regardless of being IV or OOV queries, STD is done by composing the query FSA with the index, and one can work out the posterior score, start-time and end-time from the weight of the resulting FST. In this work, we only focus on IV queries since most of the queries in our keyword lists are in-vocabulary.

5.1.2 CN Search

Our procedure for generating confusion networks is based on the Minimum Bayes Risk decoding algorithm of [48]. STD is carried out on confusion networks as follows. For single-word queries, each occurrence of the query word in the confusion networks generates a detection. The starting and ending times of the detection are those of the cluster containing the word; the score of the detection is the probability of the word. For multiple-word queries, dynamic programming is used to find all paths in the confusion networks such that the words on the path form the query. The paths may contain epsilon words. Each path generates a detection: the starting and ending times are those of the first and last clusters in the path, and the score is the product of the probabilities of all the words (including epsilon words) in the path. If multiple detections for the same query overlap, only the one with the highest score is retained.

5.2 Motivation for IR system combination

”Combination of the *different* ASR systems usually provide consistent improvement, what about combination of the *same* ASR system?” System combination is an established technique in STD. There are multiple works [21, 27] report solid improvement by combining different ASR systems. Different ASR systems can be interpreted as different decoding of the same speech signal, combining different decoding can surely yield improvement. However, the question of how to get the most information from a single decoding result remain unsolved. The goal for IR system combination is to explore the maximum potential for a single ASR system decoding result. We believe a smart way of doing retrieval can be as beneficial as an improvement on ASR system to STD task. Moreover, we believe the improvement on the IR system is additive to the existing ASR based approach. Together with the advancement in both direction, we can achieve even better on the STD task.

5.3 IR System Combination Techniques

Search results from multiple systems or different search methods are combined on a per-keyword basis. For each keyword, its detections in all the search results are pooled together. These detections are regarded as nodes of a graph; an edge is drawn between two detections if they overlap. Each connected component of this graph generates a combined detection. The starting and ending times of the combined detection are calculated as the average of those of the individual detections; the score of the combined detection is calculated with one of the following three methods [27]:

- *CombMAX*: The score of the combined detection is the maximum of the scores of the individual detections
- *CombSUM*: The score of the combined detection is the sum of the scores of the individual detections
- *CombMNZ*: The score of the combined detection is the sum of the scores of the individual detections times the number of individual detections.

In *CombSUM* and *CombMNZ*, if the resultant score is greater than 1, it is clipped to 1.

The following figure 5.1 is the IR system combination pipeline, this is the main work flow for the improved STD IR systems.

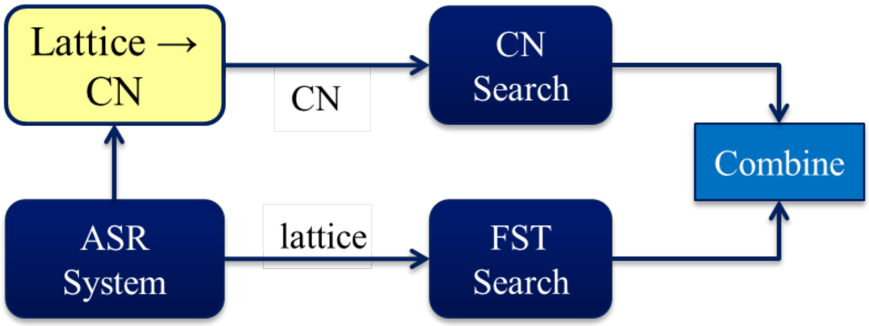

Figure 5.1: IR system combination pipeline

5.4 Dataset and experiments setup

5.4.1 Dataset

We use conversational (telephone) speech recorded in five different languages: Assamese , Bengali , Haitian , Lao and Zulu , as available in the IARPA BABEL program. For each language, there are 10 hours of training data and 10 hours of development data. We conduct our experiments using the development query sets and the development data.

5.4.2 Experiments setup

STD system description

Our STD system uses a ASR-IR two stage pipeline, which is based on the Kaldi toolkit [36]. The decoded hypotheses are represented as lattices, then converted to confusion networks. Each search apply to the respective hypotheses representation, and the detection result are combined with different IR system combination technique as presented in section 5.2.

Evaluation Metrics

Spoken Term Detection uses multiple metrics for evaluation. All metrics are based on the Term Weighted Value (TWV) [16, 47] for evaluation. The formula for TWV is as follows:

$$TWV(\theta) = 1 - (P_{\text{Miss}}(term, \theta) + \beta * P_{\text{FA}}(term, \theta))$$

where θ is the detection threshold, β is a factor that controls the balance between misses and false alarms, set to 999.9.

The concept of TWV score is simple: If the system performs perfectly on a query, it has a TWV of 1; if the system misses some of the query words or produces false alarms, it receives penalty on the TWV score. As a result, the TWV score is bounded above by 1 but has no fixed lower bound.

We use two separate metrics to describe the performance of STD systems:

- Maximum Term Weighted Value (MTWV): MTWV is the maximum TWV over the range of all possible values of the detection threshold.
- Supreme Term Weighted Value (STWV): STWV is the maximum TWV without considering false alarms. It is similar to lattice recall for a given query.

The metrics are computed on a per-query basis, and then averaged for reporting. Together these two metrics provide better insight into the overall quality for our search results, as they are not sensitive to specific detection threshold. With the appropriate detection threshold, the ATWV (The metric in last chapter) is very close to MTWV value.

Experiments description

We conducted three different sets of experiments. Each set is conducted on three different decoding systems: a Deep Neural Network (DNN) system, a Bottleneck Feature (BNF) system and a Perceptual Linear Prediction (PLP) system. Our search component only processes the IV queries, for the OOV queries, it does not output any result.

The first set of experiments compare the performance of the two different searches, FST search and CN search. The second set of experiments combine the search results from FST search and CN search to determine if we can obtain better STD performance. The final set of experiments combine all of our results to see if the gain from the individual systems is additive. The combination is also performed in different orders to note whether that affects the final result.

5.5 Experimental results

5.5.1 Comparison between FST and CN Searches

Figure 5.2 shows the MTWV for different systems on five different languages (Assamese, Bengali, Haitian, Lao and Zulu) and 3 different decoder front-ends: Deep Neural Net (DNN), Bottleneck Features (BNF), and Perceptual Linear Predictive (PLP). We performed a statistical analysis by fitting a general linear model to the data and found statistically significant differences between

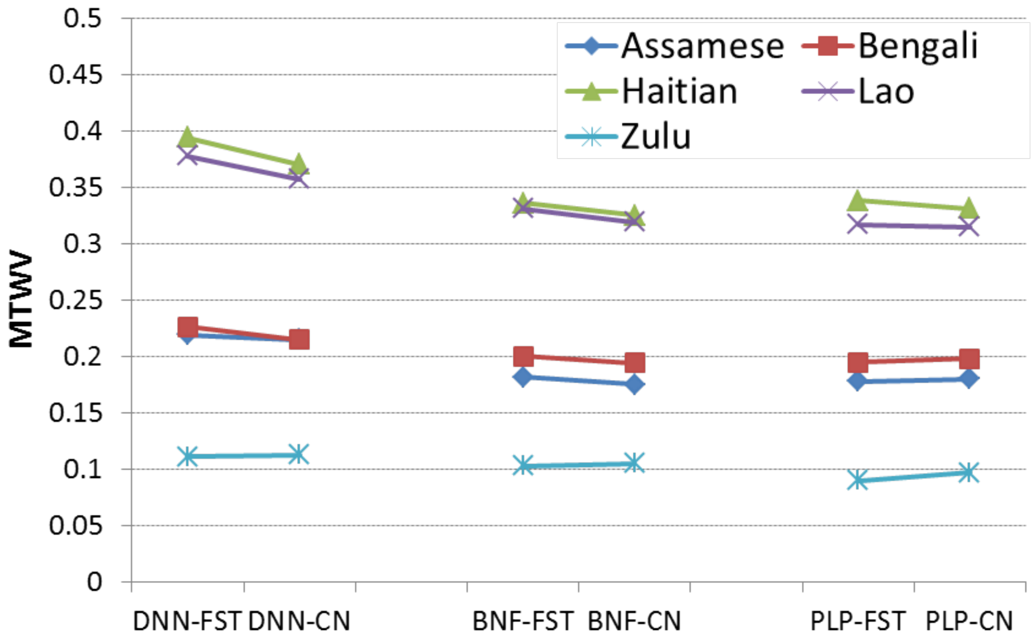

Figure 5.2: System comparison between different ASR systems and search methods.

languages, front-end features and search methods, all at p¡0.001. FST search generally outperforms CN search on every language except for Zulu. This is due to the distribution of query length (the number of word tokens per query) in the Zulu query set.

5.5.2 Combination of FST and CN searches

We evaluated three different techniques: *CombMAX*, *CombSUM* and *CombMNZ*. *CombSUM* appears to be the best way to combine FST and CN search. The results shown in Table 5.1 are averaged over front-end. It is worth noting that the performance on each decoding front-end shows the same trend as with the average performance. There are two observations that are worth making. First, the search combination has less effect on Zulu. This is due to the distribution of query length (see Table 3). Second, CN search has better performance on STWV over FST search. This is caused by the conversion from lattice to CN. The detail for both observations is discussed in the Analysis section.

5.5.3 Combination between decoding systems

The final set of experiments is carried out to determine whether the improvement from search combination is additive to the existing ASR system combinations.

After combining result from multiple searches, these results are further combined with the result from different decoding systems to achieve even greater improvement, as shown in Table 5.2. The result is the average MTWV over all languages. We pick the DNN system as our single best

Table 5.1: MTWV/STWV for search combination

Language	Metric	FST	CN	CombSUM
Assamese	MTWV	0.193	0.190	0.203
	STWV	0.369	0.372	0.380
Bengali	MTWV	0.207	0.202	0.217
	STWV	0.361	0.366	0.373
Haitian	MTWV	0.356	0.342	0.368
	STWV	0.496	0.501	0.514
Lao	MTWV	0.342	0.330	0.358
	STWV	0.474	0.476	0.492
Zulu	MTWV	0.101	0.105	0.107
	STWV	0.235	0.236	0.236

Table 5.2: MTWV/STWV from IR combination to ASR+IR system combination

Language	Metric	Single Best	IR Combination	IR+ASR Combination
Assamese	MTWV	0.219	0.229	0.248
	STWV	0.430	0.441	0.465
Bengali	MTWV	0.226	0.234	0.258
	STWV	0.407	0.417	0.445
Haitian	MTWV	0.394	0.402	0.423
	STWV	0.564	0.576	0.597
Lao	MTWV	0.378	0.396	0.418
	STWV	0.541	0.556	0.584
Zulu	MTWV	0.113	0.116	0.128
	STWV	0.264	0.265	0.279

system. By search combination, we achieve better performance on all five languages. If we combine the search combination result from other decoding systems, we gain further improvement. This indicates that the improvement from system combination comes from the diversity between systems. Although the BNF system and the PLP system have slightly worse performance compared to the DNN system, combining them nevertheless yields improvement. We also tested doing system combinations in different orders but found out that the order of combination does not have much impact on performance.

5.6 Analysis

5.6.1 Search and query length distribution

During our experiments, we discovered that the improvement from search combinations varies for different languages. On closer inspection, we found that the difference is due to the distributions of query length for each language. Each of the 5 language has around 2000 queries, yet query length distribute differently, as shown in Table 5.3.

Table 5.3: Distribution of query length in five languages

Length	Assamese	Bengali	Haitian	Lao	Zulu
1	947	926	573	325	1857
2	850	877	953	902	109
3+	162	167	398	698	19

The queries for Haitian and Lao have relatively low percentages of queries with length 1. On the other hand, the queries for Zulu have extremely high percentage of queries with length one. This distribution is highly correlated with the result showed in Table 5.1, where it is showed the search combination is more helpful for Haitian and Lao and less beneficial for Zulu. The statistical analysis indicates a significant interaction (p¡0.01) between query length and search technique.

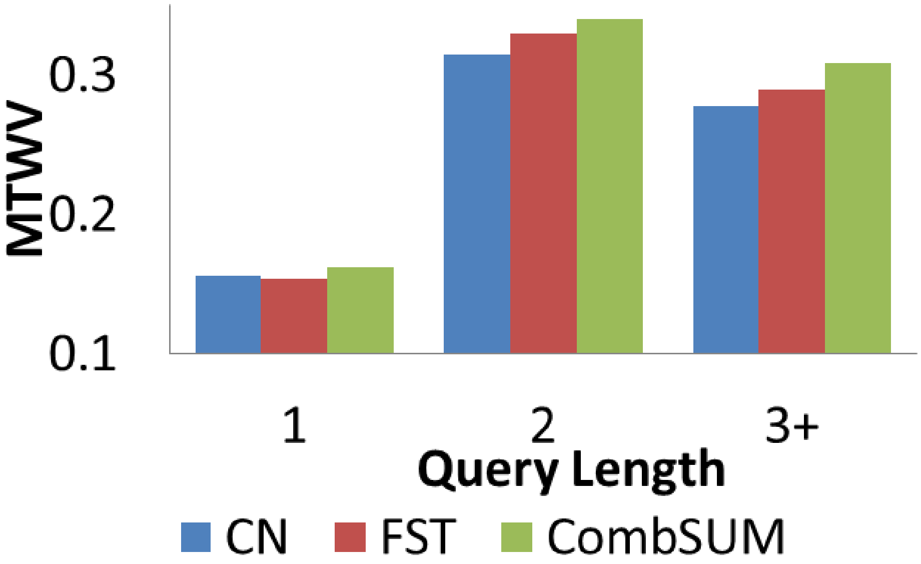

Figure 5.3: MTWV interactions for search methods and query length

Figure 5.3 shows the interactions between search methods and the query length, averaged over all languages and decoding systems. This analysis yields two findings.

CN search performs somewhat better on queries of length one word, while FST search outperforms CN search on longer queries. As well, CN search has fewer false alarms compared with FST search on the one word queries. This is a consequence of lattice to CN conversion, since hypotheses in the lattice are merged or pruned during the conversion process. The false alarm hypothesis can be pruned, or its probability can be suppressed by other well-recognized hypotheses in the same confusion set. The conversion process does not have too much impact on correct detections, since these are mostly preserved in the CN. As a result, the preserved correct detections and the removed false alarms contribute to better MTWV score. FST search outperforms CN search on multi-word queries. This is because lattices can better preserve history information for decoding hypothesis compare to CN. This observation provides an explanation for the result shown in Figure 5.2, where FST search outperform CN search on every language except for Zulu. From Table 3, we can see the query set for Zulu is mostly composed of single word queries. We believe the overall difference in MTWV is caused by the imbalanced query set, not by properties of the language.

Search combination provides better improvement on multi-word query, compared with single word query. This matches our finding that the improvement from system combinations comes from the diversity of systems. FST search and CN search use different approaches to search on multi-word queries. This diversity contributes to the consistent improvement over different languages and systems. For the single-word query, since there is less difference on two search approaches, the improvement for system combination is limited due to the lack of diversity. This answers why the search combination has less effect on Zulu. The Zulu query set is mostly single word queries, and there is insufficient diversity between the two different search approaches.

5.6.2 Search and ASR systems

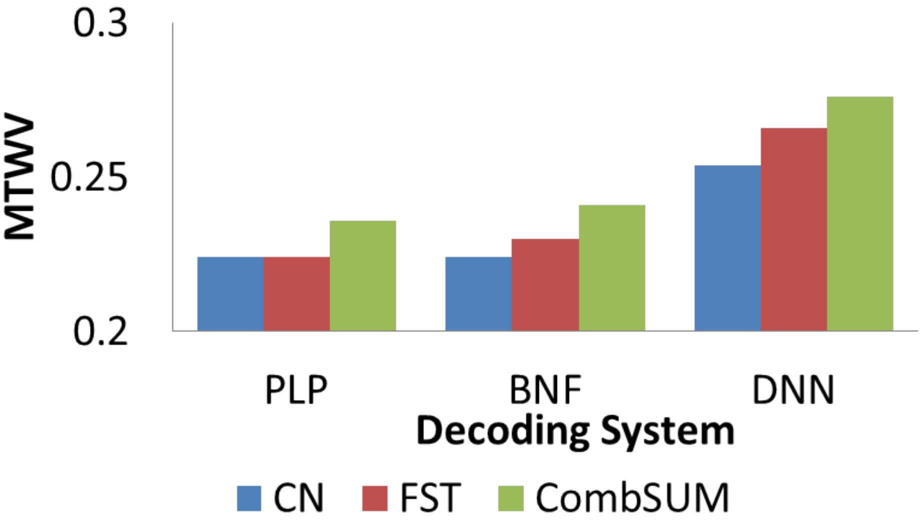

Figure 5.4: MTWV interactions for search methods and ASR systems

Figure 5.4 shows the interactions between different ASR systems and search methods. The result is the average MTWV over all languages, using different search methods. We have two observations according to this analysis. First, search combination provides consistent improvements across different decoding systems. This indicates the search combination is not sensitive to the properties of decoding systems. Second, the difference on MTWV for CN and FST search is correlated to the performance of the decoding system. The DNN system has the best overall performance on the MTWV, and the difference between FST and CN search is the largest. On the other hand, the PLP system has the worst performance on MTWV among the three systems.

The difference between FST and CN search is also the least in our experiment. This suggests that FST and CN search have similar performance on a weaker decoding result, and the difference is larger when a higher quality decoding result is available. But combining the different results can still gain extra improvement.

5.6.3 The higher STWV in CN search

From Table 5.1, we can see that CN search consistently has higher STWV compared to the FST search. This is because creation of confusion networks gives rise to extra links between words. These links are only available during CN search, and they contribute to the somewhat higher STWV. We use an example to describe this link creation process.

Figure 5.5 shows an example of link creation during confusion network conversion. In the lattice, we have two possible hypotheses AB and CD over the same time. If we use FST search, we can only find the occurrence of AB or CD. However, if we create the confusion network from the same lattice, we obtain two extra links AD and CB. This phenomenon increases the STWV for the CN system , yet does not have huge impact on the MTWV score. FST search still produces a better MTWV score over multi-word queries.

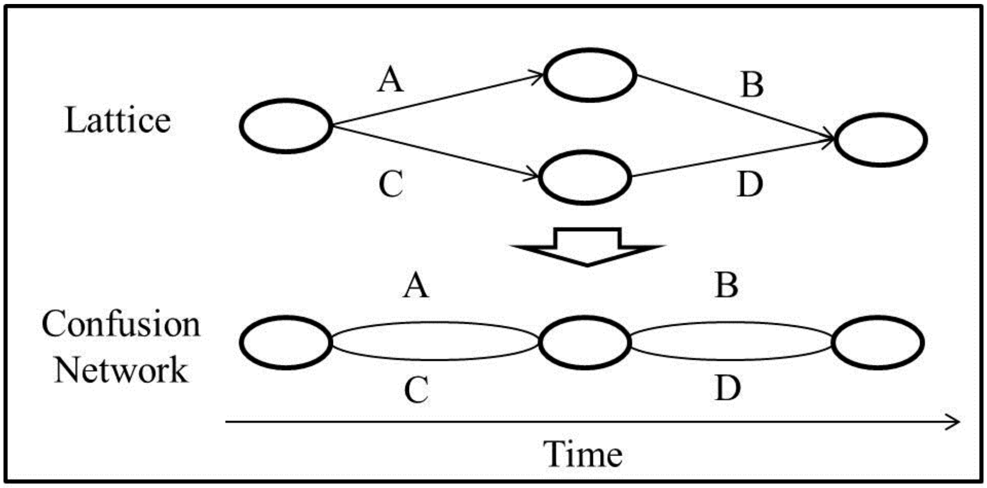

Figure 5.5: Extra link created during CN conversion

5.7 Proposed Work

5.7.1 Developing better evaluation metrics and theoretical framework

The presented work in this chapter has the same issue with the work in chapter 4, which are the lacking of objective evaluation metrics and a theoretical framework.

TWV based metrics are finely tuned metrics for specific tasks, yet it is hard to compare the performance with the other retrieval work. We propose using F-score and lattice recall to provide further insight for the current methods.

A generalized framework is helpful for understanding mathematical theory behind system combinations, and can give us insight on doing combination. After complete the proposed work here, we should be able to have better insight on what combination that will be helpful for the Speech Retreival.

5.7.2 More analysis on “How” the improvement are gained through combination

The combination presented in this chapter shows consistent improvement. It shows that both search has it’s insufficiently that need to be compensate by the other approach. We already shows one of the improvement comes from how two searches different on multi-word queries, yet there are more analysis to be done. With more analysis and better understanding of what’s the gain and lost for the search combination, we can provide the requirement for designing a better search strategy in the future.

5.7.3 Joint optimization for the IR system

We learned the combination for the existing approach can yield improvement on the IR system. Another way for refining the IR system is to perform joint optimization for the retrieval process. In this iterative process, the search algorithm changes according to the retrieved result from the last iteration. This process provide data driven way for optimizing search for different ASR results.

5.8 Summary

In this chapter, we present analysis and improvement for the current STD IR systems. We first present the two different IR system in our STD system pipeline, which are based on FST search and CN search respectively. We find that CN search performs better on single word queries, and FST search performs better on multiple word queries. We also present an improved IR system setup, which is based on the IR system combination. The combination leads to better STD results, without additional decoding. If we add extra decoding results, we can provide additive improvement on the existing STD result. We propose to develop better evaluation metric and theoretical framework for better understanding of the improvement we achieved. Also, we will do more analysis on “*how*” the improvement are gained from the combination, in order to figure out the essential element for a good IR system for Speech Retreival task.

Chapter 6

Speech Retrieval under Open Domain Conditions

In this chapter, we propose to develop Speech Retrieval systems under open domain condition. We first present the motivation for this line of work. We believe open domain data is the next stage for Speech Retrieval, after the current research on the conversational telephone speech. After the motivation, we present our target for retrieval task. Instead of a complete spoken document or a short spoken term, we focus on retrieving spoken snippet, medium size of speech which contains the essential context for the query, yet still relatively short for easier access. We then propose two different Spoken Snippet Retrieval (SSR) approaches, which are based on existing STD and SDR researches. We propose to apply the approaches we presented in previous chapter on SSR, which we expect to expand the improvement from limited resources condition to open domain condition.

6.1 Motivations

The amount of speech data available is growing rapidly in recent years. There are lots of new speech data being upload to on-line storage service such as Youtube or Vine everyday. As a result, an effective retrieval method for these data is essential for accessing the content of these data efficiently. In nowadays, most of the retrieval methods for these data are based on the title or description given by the user. This approach has its limitation for the data collected by the device that is not designed for typing text, such as Google glass. Also, the retrieved data must be in adequate size for easy access, since retrieving a few minutes of speech is hard for user to listen through the entire spoken document. As a result, we propose to develop a Speech Retrieval system which has the following properties:

- Capable for retrieving speech data recorded in different environment and different device, which we define as open domain condition
- Instead of retrieval based on text title and description, use the content as the retrieval target
- Retrieving medium size of speech which contains the essential context for the query

The open domain is the first and the most distant difference from the previous speech retrieval research. Traditional speech retrieval research mostly does not have any domain mismatch. They exploit ASR system trained from conversational speech data for speech retrieval on the same type of data. As a result, most ASR results have good WER, which makes the speech retrieval task easier. In reality, there is no guarantee that which kind of data need to be retrieved. Hence, it is not possible to prepare domain match ASR systems for the retrieval task. The domain mismatch leads to generally higher WER compare to the domain match setup, yet we believe it is more valuable to create a speech retrieval system which can perform well under domain mismatch.

The current retrieval for on-line speech collection are mostly based on the provided title or description for the data. Since the ASR result for speech data are usually noisy, using title and description as correct source of information seems a reasonable solution. However, as the new recording devices such as Google glass being invented, there will be more and more data without any text description, since not all the devices are supported with text input. Our goal is to do retrieval based on the content of the speech data, so it does not require any extra text information.

In the traditional SDR task, the average length of a spoken document is three minutes. It is time consuming for user to find the desired results within the spoken document, since unlike text document which user can skim through to find the important paragraph, spoken document must be listen through the entire document to get the content. On the other hand, most of the STD task focus on detecting the presence of a specific query term or phrase, it is also not very meaningful to only notice the location of a specific word. The presence of a word is meaningful when it comes with appropriate contexts. As a result, we propose Spoken Snippet Retrieval (SSR) to retrieve result with essence context for the query. The detail for SSR is introduced in the next section.

6.2 Spoken Snippet Retrieval (SSR) under open domain condition

The goal for Spoken Snippet Retrieval (SSR) is to retrieve a moderate size of speech from the speech collection with just enough context. We aim to retrieve snippet of speech which is related to the given query yet shorter than 20 seconds. In this case, user can understand the most essential content for the query among the speech collection, without listening through the entire spoken document. The query in the SSR task is keyword or phrases, which is similar to the query used in the STD task. The reason why we do not use natural language query as SDR task is that, it is less likely for a user to type in a complete natural language query sentence.

The open domain conditions brings the major challenge in this task. Unlike previous SDR works, which most systems have satisfactory WER from ASR systems, ASR in open domain condition is way more noisy. [25] reported ASR using large scale DNN and semi-supervised training on Youtube data has around 40% WER, which is way higher than most of the work on SDR task. How to effectively retrieve snippet from the noisy decoding result is another challenge for our task. Aside from the domain mismatch, how to select the appropriate size of snippet from the entire spoken document is another question we aim to answer.

6.3 Dataset and experiments setup

6.3.1 Dataset

We downloaded 60 hours of Youtube "How to" video as our testing data. All of the Youtube video we downloaded are provide with human transcription, so we can compute the WER for our ASR systems. Note that this dataset is easier than regular Youtube video, since "How to" video tends to speak slower and clearer for better understanding. Yet we believe this is a resonable starting point for the open domain data.

6.3.2 Experiments setup

SSR system description

Our SSR system uses a ASR-IR two stage pipeline, which is based on the Kaldi toolkit [36]. Acoustic models are trained on the Wall Street Journal corpus which consists of approximately 80 hours of broadcast news speech. We build GMM models using the Kaldi toolkit. Speaker adaptive training (SAT) is conducted via feature-space MLLR on top of LDA+MLLT features. DNN inputs include spliced fMLLR features. All decoding runs use a trigram language model which is constructed from 480 hours of YouTube transcripts. These LM training transcripts have no overlap with the testing set. The decoded testing data has 44% WER with the GMM trained acoustic model, and 39% WER with DNN trained acoustic model, which is about the same level as reported in [25]. Both one-best hypotheses and lattices are used for snippet retrieval with different retrieval system. We propose to have two distinct retrieval systems, which based on SDR and STD strategy, respectively. Further process are focus on the retrieval system instead of the ASR system.

Evaluation Metrics

We propose to use average query F-score as the evaluation metric. F-score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct results divided by the number of all returned results and r is the number of correct results divided by the number of results that should have been returned. The F-score can be interpreted as a weighted average of the precision and recall, where an F-score reaches its best value at 1 and worst score at 0. We average the F-score for each query in the entire query set to evaluate the performance for the SSR system.

The formula for F-score F is as follows:

$$F = \frac{2 * p * r}{p + r}$$

6.4 Proposed Work

We propose to complete three sets of experiments. Each set is focus on answering different research questions for SSR task. The first set of experiments compares the SSR performance with two different baseline system, which uses SDR based approach and STD based approach. This comparison shows which set up is more ideal for the further SSR development. We also plan to analysis the result from two different baseline, which give us better understanding for two search approach. The second set of experiments focus on including context from conversation for better SSR performance. We already showed context from conversations can contribute to better Speech Retrieval under limited resources condition, we plan to extend the effect to the open domain condition. Last but not least, further refinement and improvement on the SSR IR system are also planned. After analysing the experiments from the first set, we expect to have better understanding of the IR system in SSR task. We propose to present better design on the SSR IR system in order to achieve better performance.

6.4.1 SSR with SDR or STD approaches

We propose SSR as a new type of Speech Retrieval task, yet it is possible to build SSR systems based on previous approaches in SDR or STD researches. From the SDR perspective, we can build a regular SDR pipeline to retrieve possible spoken documents for the given query. After retrieved the high possibility spoken documents, we then extract essential snippet from the spoken documents as the SSR system output. STD based approach first uses regular STD pipeline to detect the presence of query words. When successfully locate the query word, the system then output the closet snippet from the detected term.

These two systems are the baseline for the SSR researches in this chapter. They reflect different strategy on SSR task, the top-down process (SDR based approach) and the bottom-up process (STD based approach). If we want to find an appropriate snippet, should we start from finding spoken documents then select the essential part to output? Or should we first detect the location of the query word, and then output some amount of context with the detected query word? These baseline integrate the efforts in Speech Retrieval field in past decades.

6.4.2 Applying context from conversation and IR improvement for SSR

In previous two chpaters, we already showed the context from conversation and the improvment on the IR system can be beneficial for Speech Retrieval under limited resources condition. We propose to develope similar techique for SSR task.

We wish to answer the question: Can these approaches provide improvement on Speech Retrieval when the ASR systems has limited performance? (either from the limited training data or domain mismatch) We propose to develop Speech Retrieval technique that can have good performance even when the ASR output is not clean and ideal.

Chapter 7

Time Line

7.1 To-do list

7.1.1 Chapter 4

- Evaluation existing results with F-score.
- Designing theoretical framework for Local rescoring and Word Burst rescoring.

7.1.2 Chapter 5

- Analysis the relations between FST and CN search on single word query and compare how two search approaches process same multi word query.
- Evaluation existing results with F-score.
- Designing Joint optimization procedure for IR system.
- Designing theoretical framework for IR combination.

7.1.3 Chapter 6

- Complete SSR baseline systems, with SDR based approach and STD based approach, and evaluate with F-score.
- Applying context from conversation for SSR systems.
- Analysis the performance on two baseline systems (SDR based approach and STD based approach) and perform combination based on the same ASR result.
- Present the more generalized theoretical framework for the work under open domain condition.

Table 7.1: Current progress

	Chapter 4	Chapter 5	Chapter 6
Baseline (20%)	✓	✓	✓/✗
Proposed Method (40%)	✓	✓	✗
Evaluation (10%)	✓	✓	✗
Analysis (15%)	✓	✓/✗	✗
New Evaluation Metric (5%)	✗	✗	✗
Theoretical Framework (10%)	✗	✗	✗
Progress (%)	85	80	10

Table 7.2: Proposed Time Line

Open domain baseline	07/2014
Open domain proposed approaches	08/2014 to 09/2014
Analysis and new evaluation metrics	10/2014
Theoretical framework	11/2014 to 12/2014
Thesis writing	01/2015 to 05/2015
Thesis defend	06/2015

Bibliography

[1] Alexandre Allauzen. Error detection in confusion network. In *INTERSPEECH*, pages 1749–1752, 2007. 3.2.2

[2] Satanjeev Banerjee and Alexander I Rudnicky. Using simple speech–based features to detect the state of a meeting and the roles of the meeting participants. 2004. 4.2.1

[3] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, pages 1554–1563, 1966. 2.2.2

[4] Leonard E Baum, JA Eagon, et al. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3):360–363, 1967. 2.2.2

[5] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, pages 164–171, 1970. 2.2.2

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 4.5.3

[7] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4): 467–479, 1992. 4.5.3

[8] Dogan Can and Murat Saraclar. Lattice indexing for spoken term detection. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(8):2338–2347, 2011. 5.1.1

[9] Guoguo Chen, Sanjeev Khudanpur, Daniel Povey, Jan Trmal, David Yarowsky, and Oguz Yilmaz. Quantifying the value of pronunciation lexicons for keyword search in lowresource languages. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8560–8564. IEEE, 2013. 5.1.1

[10] Guoguo Chen, Oguz Yilmaz, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. Using proxies for OOV keywords in the keyword search task. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 416–421. IEEE, 2013. 2.3.2, 5.1.1

[11] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5644–5647. IEEE, 2011. 3.2.2

[12] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012. 3.2.2

[13] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980. 2.2.1

[14] Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977. 2.2.2

[15] Michael Finke, Petra Geutner, Hermann Hild, Thomas Kemp, Klaus Ries, and Martin Westphal. The karlsruhe-verbmobil speech recognition engine. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 83–86. IEEE, 1997. 4.3.2

[16] Jonathan G Fiscus, Jerome Ajot, John S Garofolo, and George Doddingtion. Results of the 2006 spoken term detection evaluation. In *Proc. SIGIR*, volume 7, pages 51–57. Citeseer, 2007. 1.1, 3.2, 4.3.2, 5.4.2

[17] John S Garofolo, Cedric GP Auzanne, and Ellen M Voorhees. The TREC Spoken Document Retrieval Track: A Success Story. *NIST SPECIAL PUBLICATION SP*, (246):107–130, 2000. 1.1

[18] Jean-Luc Gauvain, Yannick de Kercadio, Lori Lamel, and Gilles Adda. The LIMSI SDR System for TREC-8. In *TREC*, 1999. 3.1

[19] Frederick Jelinek, Bernard Merialdo, Salim Roukos, and Martin Strauss. A Dynamic Language Model for Speech Recognition. In *HLT*, volume 91, pages 293–295, 1991. 3.3

[20] Sue E Johnson, Philip C Woodland, Karen Sparck Jones, and P Jourlin. Spoken Document Retrieval for TREC-8 at Cambridge University. In *TREC*, 1999. 2.3.1, 3.1

[21] Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, Le Zhang, Shivesh Ranjan, Tim Ng, Roger Hsiao, Guruprasad Saikumar, Ivan Bulyko, Long Nguyen, et al. Score normalization and system combination for improved keyword spotting. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 210–215. IEEE, 2013. 3.2.2, 5.2

[22] KM Knill, MJF Gales, SP Rath, PC Woodland, C Zhang, and S-X Zhang. Investigation of multilingual deep neural networks for spoken term detection. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 138–143. IEEE, 2013. 3.2.2

[23] Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(6):570–583, 1990. 3.3

[24] Julien Kupiec. Probabilistic models of short and long distance word dependencies in running text. In *Proceedings of the workshop on Speech and Natural Language*, pages 290–295. Association for Computational Linguistics, 1989. 3.3

[25] Hank Liao, Erik McDermott, and Andrew Senior. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 368–373. IEEE, 2013. 6.2, 6.3.2

[26] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. Vocabulary independent spoken term detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–622. ACM, 2007. 3.2.1

[27] Jonathan Mamou, Jia Cui, Xiaodong Cui, Mark JF Gales, Brian Kingsbury, Kate Knill, Lidia Mangu, David Nolden, Michael Picheny, Bhuvana Ramabhadran, et al. System combination and score normalization for spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8272–8276. IEEE, 2013. 3.2.2, 5.2, 5.3

[28] Lidia Mangu and Mukund Padmanabhan. Error corrective mechanisms for speech recognition. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 29–32. IEEE, 2001. 3.2.2

[29] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus among words: lattice-based word error minimization. In *Eurospeech*. Citeseer, 1999. 2.2.3

[30] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400, 2000. 2.2.3

[31] Lidia Mangu, Hagen Soltau, Hong-Kwang Kuo, Brian Kingsbury, and George Saon. Exploiting diversity for spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8282–8286. IEEE, 2013. 3.2.2

[32] Yajie Miao and Florian Metze. Improving low-resource CD-DNNHMM using dropout and multilingual DNN training. In *Proc. Interspeech*, pages 2237–2241, 2013. 3.2.2

[33] Yajie Miao, Florian Metze, and Shourabh Rawat. Deep maxout networks for low-resource speech recognition. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 398–403. IEEE, 2013. 3.2.2

[34] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish. Rapid and accurate spoken term detection. In *INTERSPEECH*, pages 314–317, 2007. 3.2.1

[35] Stefan Ortmanns, Hermann Ney, and Xavier Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech & Language*, 11(1):43–72, 1997. 2.2.3

[36] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In *Proc. ASRU*, pages 1–4, 2011. 5.1.1, 5.4.2, 6.3.2

[37] Daniel Povey, Mirko Hannemann, Gilles Boulianne, Lukas Burget, Arnab Ghoshal, Milos Janda, Martin Karafiát, Stefan Kombrink, Petr Motlicek, Yanmin Qian, et al. Generat-

ing exact lattices in the WFST framework. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4213–4216. IEEE, 2012. 5.1.1

[38] Lawrence R Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993. 2.2.1

[39] Ronald Rosenfeld and Xuedong Huang. Improvements in stochastic language modeling. In *Proceedings of the workshop on Speech and Natural Language*, pages 107–111. Association for Computational Linguistics, 1992. 3.3

[40] M Siegler, Rong Jin, and Alexander G Hauptmann. CMU Spoken Document Retrieval in TREC-8: Analysis of the role of Term Frequency TF. In *TREC*, 1999. 3.1

[41] Amit Singhal and Fernando Pereira. Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41. ACM, 1999. 3.1

[42] Amit Singhal, Steven P Abney, Michiel Bacchiani, Michael Collins, Donald Hindle, and Fernando CN Pereira. AT&T at TREC-8. In *TREC*, 1999. 3.1

[43] Amit Singhal, John Choi, Donald Hindle, David D Lewis, and Fernando Pereira. At&t at TREC-7. *NIST SPECIAL PUBLICATION SP*, pages 239–252, 1999. 3.1

[44] Svetlana Stoyanchev, Philipp Salletmayr, Jingbo Yang, and Julia Hirschberg. Localized detection of speech recognition errors. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 25–30. IEEE, 2012. 3.2.2

[45] Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tur. Semantic Parsing Using Word Confusion Networks With Conditional Random Fields. In *Proc. of the INTERSPEECH*, 2013. 3.2.2

[46] Dimitra Vergyri, Izhak Shafran, Andreas Stolcke, Venkata Ramana Rao Gadde, Murat Akbacak, Brian Roark, and Wen Wang. The SRI/OGI 2006 spoken term detection system. In *INTERSPEECH*, pages 2393–2396. Citeseer, 2007. 3.2.1

[47] Steven Wegmann, Arlo Faria, Adam Janin, Korbinian Riedhammer, and Nelson Morgan. The TAO of ATWV: Probing the mysteries of keyword search performance. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 192–197. IEEE, 2013. 4.3.2, 5.4.2

[48] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu. Minimum Bayes Risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language*, 25(4):802–828, 2011. 5.1.2

[49] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. *submitted to ICASSP*, 2014. 3.2.2