Features for Search and Understanding of Noisy Conversational Speech

Justin Chiu

CMU-LTI-18-005

May 2018

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

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
In Language and Information Technologies

Keywords: Features, Conversational Speech, Noisy Automatic Speech Recognition

Abstract

As the amount of speech data available increases rapidly, so does the need for efficient search and understanding. Techniques such as Spoken Term Detection (STD), which focuses on finding instances of a particular spoken word or phrase in a corpus, try to address this problem by locating the query word with the desired meaning. However, STD may not provide the desired result, if the Automatic Speech Recognition (ASR) system in the STD pipeline has limited performance, or the meaning of the item retrieved is not the one intended. In this thesis, we propose different features that can improve the performance on search and understanding of noisy conversational speech.

First, we describe a Word Burst phenomenon which leverages the structural property of conversational speech. Word Burst refers to a phenomenon in conversational speech in which particular content words tend to occur in close proximity of each other as a byproduct of the topic under discussion. We design a decoder output rescoring algorithm according to Word Burst phenomenon to refine our recognition results for better STD performance. Our rescoring algorithm significantly reduced the false alarm that were produced by the STD system. We also leverage Word Burst as a feature for identifying recognition errors in conversational speech. Our experiments show that including Word Burst feature can provide significant improvement. With this feature, we demonstrate that higher level information, such as structural property can improve search and understanding without the need for language-specific resources or external knowledge.

Second, we identify the mismatch between different decoder output created by the same ASR system can be leveraged as a feature for better STD performance. After the decoding process of an ASR system, the result can be stored in the format of lattice or confusion networks. The lattice has richer historical information for each word, while the confusion network maintain a simple and more compact format. Each of this format contains unique information that is not presented in the other format. By combining the STD result generated from these two decoder output, we can achieve improvement on STD systems as well. This feature shows that unexplored information could be stored in different output generated by the identical ASR system.

Last but not least, we presented a feature based on distributed representations of spoken utterances. Distributed representations group similar words closer in a vector space according to its context. Every word that shows up in a regular context will be projected into the vector space closely to each other. As a feature space, we not only project the word in the space, but also project the utterances that contains multiple words into the space. We apply this feature to Spoken Word Sense Induction (SWSI) task, which differentiates target keyword instances by clustering according to context. We compare this approach with several existing approaches and shows that it achieves the best performance, regardless of the ASR quality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Conversation is one of the main methods that humans use to communicate with each other. In recent years, products that use conversational interfaces have appeared and have allowed humans to communicate with smart devices or computer , such as Siri, Google Home or Amazon Echo. As the technology keeps progressing, we believe there will be more computer systems or devices interacting with human through conversational speech. As a result, being able to properly understand such speech become a critical task for building systems that communicate through language.

An inherent issue in processing conversational speech is that it is imprecise in several ways. When the speech is being produced in an informal situation, it will not always be well structured. For example, when chatting with friends, a spoken utterance might not be grammatically correct but it will nevertheless make sense. Similarly, when speech is been produced in difficult environments, background noises might make the speech harder to make out. It will affect the performance of the Automatic Speech Recognition (ASR) system. In addition to imprecision and noisiness, there are other challenges, conversational interfaces are also expanding to different languages that might not have as many resources such as English. These conditions constitute challenges for speech understanding.

A key motivation for our work is to identify potential features or approaches that can provide improvement to speech processing in addition to traditional modeling techniques, that are effective regardless of a particular language, and focus more on the properties and structure of conversation that derive from its communicative functions. We believe that it is possible to provide general improvement to speech understanding as a whole, using techniques based on how people use generic knowledge of conversational structure to improve their understanding in different situations.

1.1.1 The challenge of existing approaches

There are many contemporary technologies that can be used to perform search and understanding of speech. Most of these use Machine Learning models to conduct their analysis. However,

Machine Learning based approaches have some challenges:

- Robustness: Machine Learning models tend to perform well on clean data, as the features extracted from data can be more discriminative. When the data are noisy, the quality of extracted features is affected and hence impairs knowledge extraction performance. For example, Automatic Speech Recognition (ASR) on clean speech corpus such as Switchboard can achieve the WER under 20% (Hinton et al., 2012), while ASR on noisy corpus still have a WER over 50% (Miao and Metze, 2013). This performance difference shows the challenge on the robustness aspect. The reason for this difference is because the features extracted from the clean data can fit to the trained model better, while the features extracted from noisier data contains more noise and can not fit to the model that well. Noise will be an issue because the data we collect nowadays will come from different environments and hard to ensure it is always clean data.
- Training data: High-performance Machine Learning models tend to require significant amounts of data to train a good model. The data also requires human labels/knowledge to make them usable for training. Collecting a significant amount of human-labeled data for training is expensive, especially data that is difficult to obtain, e.g. languages that are used in less developed regions, where Internet is not available.
- Portability: There are approaches that utilize language-specific features such as tone in Chinese for Automatic Speech Recognition (ASR) (Fu et al., 1998). These kinds of approaches can achieve impressive improvement on a single language yet lack portability to adapt to other languages, and features like tones might not even exist in some other languages.

To demonstrate the challenge of existing approaches, an example based on Spoken Term Detection (STD) task will be provide in section 1.3.1, after its task introduction.

1.2 Thesis Statement

Humans communicate with each other through conversation in order to exchange information and knowledge. Despite the fact that our daily conversation is noisy, we are still able to receive the information without too much difficulty. We think the reason for this is because humans expect conversation with other people to happen in certain ways. There are multiple phenomena that can be observed in human conversations and that can be treated as a feature to support human understanding of other people's speech. First, words that has been spoken recently in conversation is more likely to recur in close proximity. We refer to those as the Word Burst phenomenon in this thesis. Second, if an identical word had been spoken in the conversation with very similar context, it tends to have similar meaning. However, if the same word had been spoken in a different context, its meaning is quite possibly different. These phenomena reflect human's communicating knowledge and delivering semantics through conversations. Since humans are able to leverage these phenomena to support understanding, we anticipate that it should also be beneficial for automatic systems. Aside from these features from the conversational side, we also identify another feature from the automatic system side. When an automatic system processes its input, the system can store the processed result in different representations. Each

representation has its own strengths and weaknesses, and the mismatch between different representations contains information that can provide more accurate at expectation the original data. In this thesis, we aim to leverage these phenomena to support search and understanding of speech collections. Our approach can also address challenges found in existing approaches to automatic speech recognition:

- Robustness: Since the data we process are noisy, we need to use additional information in order to improve the quality of input data for Machine Learning models. The phenomena we aim to leverage serve this purpose.
- Training data: The approaches we propose do not require specific training data, as they either depend on phenomena related to how humans organize their words in conversation or how automatic systems represent these results.
- Portability: Since we are leveraging phenomena from either generic human conversation or generic automatic systems, our approaches are unlikely to have any language-specific limitations.

One important note we wish to clarify is that, for the two conversational oriented phenomena that we leverage, we are not leveraging "semantics". The information we leverage can be considered as a feature of semantics, as the reason for its existence is to support the exchange of semantic information between people. It can, at most, be considered as an observable manifestation of semantics. We believe semantic is an abstract concept and the definition of it is subjective. However, the feature we leveraged in the thesis is an objective phenomena that can be utilized consistently among different experiments.

To be more precise, the research question we identified in existing work is the challenge we described in last section: existing approaches might not be robust enough when training data is insufficient. Hence, this also limits the portability of the approach to language with insufficient data. We propose to identify high level features in human conversation that enhance the search and understanding of noisy conversational speech. The features we explore will address the challenge of existing approaches since it does not require large amounts of training data nor language specific features, and can be ported to different languages easily. This should support the development of search and understanding systems for human speech in various languages, especially languages with insufficient data available. In addition, this will also improve our understanding of conversational activities, which is a great source of language-independent features for understanding human speech.

1.3 The tasks

In this thesis, we work on three different tasks, Spoken Term Detection (STD), Identifying Recognition Errors, and Spoken Word Sense Induction (SWSI). We will first introduce the three tasks, their current challenges, and how we are going to address them. After the tasks introduction, we will describe our motivation for selecting these tasks as the focus for the thesis.

1.3.1 Spoken Term Detection

Task Introduction

Spoken Term Detection (STD) focuses on finding instances of a particular spoken word or phrase in an audio corpus. It is also called Key Word Search (KWS), yet we will unify the naming as STD in this thesis. The STD system inputs a set of text queries, and it should output the locations of the given text query in the audio corpus. It was proposed as a task in order to efficiently retrieve information from a growing body of computer-accessible volumes of audio data.

Most STD is accomplished by using combinations of Automatic Speech Recognition (ASR) systems and term searching systems. An ASR system is applied to an audio stream and generates a time-marked recognition hypothesis of the speech. The performance of an ASR system is evaluated by Word Error Rate (WER), which is the word level Levenshtein distance between the decoded word and the ground truth. For the term searching purpose, sometimes the ASR performance will also be evaluated with lattice recall, the recall value of the given query in the recognized lattice. The recognition hypothesis (which can be the single best only or the entire lattice/confusion network) is then indexed and searched by an term searching system, and the result returned for a query unit is a list of possible locations for the query unit ordered by decreasing probability.

The evaluation metric of STD is mostly related to a value called Term Weighted Value (TWV) (Fiscus et al., 2007; Wegmann et al., 2013). The formula for TWV is as follows:

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

Different TWV based value can be used for evaluation, including Actual Term Weighted Value (ATWV), Maximum Term Weighted Value (MTWV) or Supreme Term Weighted Value (STWV). ATWV is the average of TWV over all queries; MTWV is the maximum TWV over the range of all possible values of the detection threshold; STWV is the maximum TWV without considering false alarms. It is similar to lattice recall for a given query.

The concept of the 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 a penalty on the TWV score. As a result, the TWV score is bounded above by 1 but has no fixed lower bound.

TWV-based evaluation metrics had been reported as very unstable metrics (Wegmann et al., 2013). These metrics depend on specific parameters, and different evaluations could have different parameters. In this thesis, we use the parameters provided by the IARPA BABEL program. However, due to these parameters, the results that have been reported on different data sets might not be comparable with each other if the parameters are not the same. Only the relative comparison on the same dataset with the same parameters can be used as a way of evaluating whether the approach can provide improvement.

The most intuitive way of doing STD is to use ASR systems to decode the speech to produce a single-best recognition hypothesis and then identify whether the query term shows up on the decoded hypothesis. In this case, the term searching step becomes trivial. This works well when the ASR performance is below 20% WER (Miller et al., 2007), so in the condition when ASR system with such performance is available, STD is considered less challenging. However,

when the ASR performance is limited, identifying query terms on the single-best recognition hypothesis is not enough, and it opens up an entire research space.

Table 1.1: WER and ATWV on using single-best recognition hypothesis or lattice (from (Miller et al., 2007))

Language	**WER(%)**	**1 best (ATWV)**	**lattice (ATWV)**	**1 best ATWV / lattice ATWV (%)**
English	14.9	0.754	0.852	88.5
Chinese	31.7	0.228	0.343	66.5

(Miller et al., 2007) reported their STD performance on English and Chinese with the comparison as shown in Table 1.1. When the WER is low, using the single-best recognition hypothesis can achieve 89% performance compared with searching in the lattice. However, as the WER increases, the difference between searching on the single-best recognition hypothesis and lattice becomes more significant.

Due to this report, the ASR system is generally considered more important in STD tasks, since if the ASR has decent performance, the term searching system search becomes less challenging in STD. In addition to that, if the actual query term is not recognized at all, the term searching system still cannot retrieve it. As a result, publications in the STD domain are mostly trying to improve the recognition performance, which can be estimated by reducing the WER or increasing the lattice recall in ASR for better STD performance.

Current Challenge

The current challenge in STD is the limitation of ASR performance under certain conditions. One of the conditions that has received a significant amount of research attention recently is the Limited Resources condition. Under Limited Resources condition, the training data for the ASR system is not sufficient for creating a robust model. The amount of training data in this condition is about 10 hours of speech (Miao and Metze, 2013), which is far less than a classical ASR system that is trained with more than 100 hours of speech(Dahl et al., 2012). A good example for this condition is to perform STD on Limited Resources languages such as Tagalog or Pashto. Research on these directions also focuses on applying their technique to multiple languages, so it cannot use language-specific features to address the challenge. Even with the recent advance in Deep Neural Networks for ASR (Dahl et al., 2012), ASR under such condition still performs badly. (Miao et al., 2013) reported that their Deep Learning ASR systems have WER from 69.9% to 72.0% in multiple languages under such condition. This is far from the ASR performance reported in previous STD tasks that are not under this condition (Miller et al., 2007), in which they achieve 14.9% WER. The noisy ASR result significantly degrades STD performance. This huge gap not only happens on WER but also on ATWV value. In 2007, (Miller et al., 2007) reported that their STD performance on English recordings reached 0.852 Actual Term Weighted Value (ATWV). However, the same team reported their STD performance on Pashto recordings in 2013 (Karakos et al., 2013). Even though the technology had advanced for six years, their best performance was 0.492 ATWV. When the challenge we described was tested, the performance of the system degraded significantly.

Our Approaches for the Challenge

On the Limited Resources STD task, we present two different approach to improve performance. Note that both of these approaches are focused on the term searching systems, as we believe this can provide improvements in addition to the progress of the ASR systems. We think the research on STD should not overemphasize the ASR system, and our work here demonstrates that improving term searching systems can also achieve reasonable improvement on STD tasks.

First, we describe two different ways to rescore the recognition hypothesis in conversational speech, Word Burst rescoring (Chiu and Rudnicky, 2013) and Unique Penalization rescoring, both relying on the Word Burst phenomenon, which we consider as a feature in conversational speech. Word Burst describes the phenomenon in conversational speech in which particular content words tend to occur in close proximity of each other as a byproduct of the topic under discussion. Based on this phenomenon, we present Word Burst rescoring, a rescoring algorithm that focus on complicated rescoring within a smaller temporal window, and Unique Penalization, which focuses more on identifying the proper window size in conversational setup for the Word Burst phenomenon. In Chapter 4, we describe this work in detail.

Second, we identify the mismatch between different recognition hypothesis created by the same ASR system can be leveraged as a feature for better STD performance (Chiu et al., 2014). The intuition of this work is as follows: When we ask a single person to present the "same" information to different people, the person will structure their communication according to their audience. These different forms of communication will have their unique and missing information about the original raw information, and it will be the same for different decoder output generated by the same ASR system. If we can leverage the the differences of multiple representations generated from identical input data, we can acquire more information about the original data. The detail of this work is described in Chapter 5.

1.3.2 Identifying Recognition Errors

Task Introduction

Works had been done in the Confidence Measures (CM) (Jiang, 2005) domain focusing on evaluating the reliability of decoder output. Researchers have proposed computing a score that indicates the reliability of a hypothesis. We propose to simplify this task to a binary classification problem; each word generated from the ASR system can be classified into one of two classes: correctly recognized or recognition error.

This simplification provides a clearer separation between decoder output. The advantage of this separation is to make it easier to evaluate the performance based on additional features, such as applying Word Burst as feature. Every word will be assigned to a label or correct or error. For example, in the CM setting if a specific word is close to a rejection threshold, it is hard to decide whether that word is recognized correctly or not. In our framework, there would be less ambiguity. This makes the application of CM result easier, as it removes the need of identifying the threshold for CM scores. In addition to that, instead of focusing on ASR system to address the CM task, this approach attempt to leverage the information that belongs to the conversation, which is a new source of information that can potentially be utilized.

Current Challenge

Most of the work on CM focuses on leveraging the information from ASR system. The situation of how the recording is created is not considered as a focus for addressing this problem. However, if we already knows that the decoding target is conversational speech, there are information that lies in conversation that can be leveraged in supporting our task of identifying recognition errors.

Our Approaches for the Challenge

As an extended application of Word Burst phenomenon, we apply it as a feature for improving our performance. We aim to demonstrate that conversational feature can be useful for tasks other than STD. In addition, the way to leverage Word Burst in STD has been criticized by researchers in different domain for being too ad-hoc. We also identify a more systematic way of leveraging Word Burst phenomenon in this task. The detail of the work will be describe in the Chpater 4.

1.3.3 Spoken Word Sense Induction

Task Introduction

Word Sense Disambiguation (WSD) is the task of identifying which sense of a word is used in a statement, when the word has multiple meanings. Many approaches have been proposed to address this problem, ranging from dictionary-based methods that use the knowledge encoded in lexical resources to supervised machine learning methods with a classifier trained having a sense-annotated corpus. (Yarowsky, 1995) Word Sense Induction (WSI) (Navigli, 2009) addresses the same problem, except that WSI does not require any external resources such as dictionaries or sense-annotated data, because it aims towards data driven approach. As a result, WSI can be considered as an unsupervised clustering problem for multi-sense words. Spoken Word Sense Induction (SWSI) enables WSI on human speech instead of natural language text. Since speech data is noisier and (spontaneous) spoken language is less structured, we anticipate a greater challenge in SWSI, compared with a text based WSI task. The state of the art in these fields is described in Section 3.3.

Current Challenge

The challenges for WSI/SWSI coming from multiple perspectives. The high-level challenge questions the fundamental nature of the problem. Is this a valid task? How are we going to evaluate it? The low-level challenge is about whether the system can maintain the performance when the context is noisy, since the context is the source of features for clustering.

From the high-level perspective, it is very difficult to define a specific word sense that belongs to a specific keyword instance. Since the word sense perceived by humans is dependent on the interpretation, there is not even a ground truth that can be accepted by every person. We believe the definition of word sense is similar to a tree-like structure, where each end node represents a word sense (Hovy et al., 2006). The deeper into the tree, the more sense (node) that is available for a specific word, yet there is no optimal method for deciding the ideal depth.

Since it is difficult to define the most appropriate ground truth, evaluating the WSI performance is another challenge. One way for evaluating the performance is to use human-transcribed word sense as ground truth, and try to map every clustered result into the ground truth. Although this operationalizes the definition of word sense, two challenges remain:

- Widely accepted evaluation metrics: A recent workshop (Navigli and Vannella, 2013) provided several evaluation metrics for reporting performance, because individual metrics have their limitations. Most metrics will be affected by the chance agreement between clusters, which makes it even harder to compare between different numbers of assigned clusters. (As the number of clusters is different, the chance agreement is also different.) The lack of standardized evaluation metric will make the field difficult to compare the work produced by different groups.
- How to map the generated cluster to specific word senses: Since SWSI places a word into multiple clusters according ot the word sense, the mapping between the generated cluster to specific word senses is another challenge. (Navigli and Vannella, 2013) uses an ideal scenario, where the generated cluster and the word sense always maximize the cardinality of the intersection. However, this is not the case for real-world applications.

From a more practical perspective, when performing clustering for our target term, can we maintain the performance when the data is noisy? Is there a way of identifying a more robust feature for the clustering? Since we target at SWSI, the noise in the context is inevitable, as the noise can come from the recognition error from the ASR system or spontaneous spoken language. As a result, we need to investigate real world noisy data to understand how it affects our approach.

Our Approaches for the Challenge

For the two challenges of widely accepted evaluation metrics and mapping between the generated cluster to specific word sense, we select the evaluation metric described by (Hubert and Arabie, 1985) that is not affected by chance agreement, and also does not require the mapping between the generated cluster and a specific word sense, as the evaluation is based on the distribution of clusters. We understand that we avoid those challenges instead of solving them, yet solving those problems is beyond the scope of this thesis.

The low-level challenge is the main challenge we address in this thesis. We design a new way of representing the context of a target term according to our observations for conversational speech, that words that occur in similar contexts tend to have similar meanings. We also apply the other part of the observation, which is that the same word occurring in very different contexts is less likely to have the same meaning. We use the first part to create a word embedding space to represent the relationship between different words, and the second part to separate the meaning of identical words. The detail of this work is described in Chapter 6.

1.3.4 Motivation for selecting the tasks

There are several reasons why we select these particular tasks in this investigation.

First, these tasks focus on spoken data. Speech is one of the most common methods of

communication for most people. And the most common form of communication is the best target to start with.

Second, these tasks can address the challenge of existing approaches that we presented in Section 1.1.1. Since these tasks have to process noisy data, this ensures that our approach can be robust to noise. Limited Resources STD and identifying recognition errors have only a very limited body of data available, and SWSI is not expect to use any labeled data, so our approach will not require too much training data, which is the second challenge. In fact, our approach on STD requires no training data. And since all of our approaches do not require training data, this also makes the approaches' portability to different languages more likely.

Third, these tasks focus on the smallest unit of our daily communication: the word. This can avoid the situation where the approach provides improvement on larger units of communication, yet the effect cannot be extended to the smallest unit. For example, a good retrieval model can improve the retrieval performance on a larger segment (60 seconds or more) (Chiu and Rudnicky, 2014) of speech data, but it has limited improvement on identifying whether a specific word's presence is correctly identified or not.

Fourth, these tasks are focused on the problem of extracting or identifying useful information from very large speech data sets, which fits the real-world scenario we described in the Motivation of the thesis. There are also real-world applications that can be derived from this research. For STD, identifying query terms from huge collections can also help the user to retrieve useful information from a speech corpus. Identifying recognition errors can help us process the recognized result more carefully. Once the relevant instances are retrieved from the corpus, SWSI can cluster the retrieved result into different groups in order to let the user access it more easily.

1.4 Summary of Thesis Contribution

This thesis investigates multiple phenomena in conversational speech and in automatic systems that can be leveraged to support automatic systems to more effectively extract useful information from very large speech collections. Overall, the most important contribution is that we identify and apply these feature that are robust to noise, require no training data, and can easily adapt to different languages. We show that our results generalize over three different tasks, STD, identifying recognition errors, and SWSI. These features include Word Burst phenomenon, mismatch between recognition hypothesis in different format, and Distributed Representations of Utterances. Based on these features, several contributions are made:

In Chapter 4, we describe Word Burst rescoring and Unique Penalization rescoring, two algorithms based on the Word Burst phenomenon. The critical feature we leverage is: A word that already appears in a conversation is more likely to recur in close proximity, and the word that appears alone without other identical words around is more likely to be a recognition error. We designed rescoring algorithms to refine our ASR results to achieve better STD performance on Limited Resource languages. We tested on six different datasets in different Limited Resource languages, and our approach achieves significant improvements, which indicates that this phenomenon can be observed in multiple languages. After the STD experiments, we leveraged Word Burst as a language independent feature on identifying recognition errors. The experiments is conducted on multiple languages and different ASR quality. We demonstrate that, even without

parameter tuning on development data, Word Burst can be used to improve the identification of potential recognition errors. The only limitation of the approach can be attributed to the characteristics of the data, which we will also describe in the chapter.

In Chapter 5, we describe a strategy of integrating multiple noisy decoder output from the same ASR system to improve STD performance. The key feature in this work is that different forms of decoder output will have their unique and missing information about the original information. If we can combine these different decoder output, we can leverage the feature of mismatch between different decoder output by the STD systems. We tested on five different datasets in different languages and the decoder output generated from three different ASR systems, and our approach can achieve significant improvements, which indicates that this can be leveraged on different languages and different qualities of ASR systems.

In Chapter 6, we design a novel and robust method of feature extraction for SWSI. We leverage the relationship between the word and its context according to our observation in conversational speech (when multiple identical words have been spoken in the conversation, the relationship between their meaning can be decided by their context) to create word embedding to represent the similarity between words, and then project the utterance that our word belongs to into the word embedding space to separate the meaning of identical words. Our experiments are conducted on three different levels of ASR quality. We compare this approach with several existing approaches and demonstrate that it achieves the best performance, regardless of the ASR quality.

1.5 Thesis Organization

The reminder of this thesis is organized as follows:

- Chapter 2 introduces the related work for conversational features we discussed in the thesis.
- Chapter 3 describes the task-level related work, which includes ASR, STD, and SWSI.
- Chapter 4 presents the application of Word Burst phenomenon. We perform experiments on different languages, using Word Burst rescoring and Unique Penalization rescoring algorithms to improve STD performance, and Word Burst phenomenon as a feature to improve identification of recognition errors. We also provide various analyses for our approach, since the effectiveness of these approaches is sensitive to multiple factors.
- Chapter 5 presents our work on the leveraging the mismatch from a same ASR system as a feature for STD in Limited Resources languages. We find that the improvement from this approach is also independent of languages and ASR systems. We combine this approach with existing multi-ASR system combination, and find that the improvement is additive with it.
- Chapter 6 presents our work on SWSI by designing a novel and robust method of feature extraction for the context of a target term to create Distributed Representation of utterances. We compare our approach with several existing approaches and show that it achieves the best performance, regardless of the ASR quality.
- Chapter 7 presents the conclusion and the future work.

Chapter 2

Related Work for Conversational Features

In this chapter, we discuss the related work on conversational features. The approach we present in the thesis leverages two different phenomena that can be observed in human conversations. The first one is Word Burst, which is that a word that has been spoken recently in conversation is more likely to recur in close proximity, the temporal information for word occurrence is the key for this phenomena. The second one is that words that occur in similar contexts in conversation tend to have similar meanings. We discuss earlier works for these two phenomena. Although we are looking at these phenomena from different perspectives and applying it to different tasks, we believe that the core ideas are similar.

2.1 Word Burst phenomenon

2.1.1 Word Recurrence in Dialogue Systems

There are research efforts in the dialogue system domain that focus on predicting what people will say to a dialogue system. If we are able to predict what a user could say, the response of the dialogue system can be better customized. (Barnett, 1973) propose "Thematic Memory" as the model to predict what user will say in the incoming conversation. It is assumed that the user will exhibit goal-directed behavior toward finding specific information relating to a universe that is small compared with the every information that are available. Content words (item names and values) contained in the most recent questions and answers are retained by the thematic memory and proposed as highly likely to occur in the next utterance.

This work starts to identify the phenomenon that when a content word is spoken, it is more likely to occur in close proximity of the same content word in a conversation. This phenomenon is leveraged as a source of information for deciding the strategy for a dialogue system in this work. Compare to how we leverage Word Burst, it focuses more on the turn taking information (recent question, which is the last turn), not leveraging the temporal information which we believe is useful.

2.1.2 Word Recurrence in Automatic Speech Recognition (ASR) Systems

(Young et al., 1989) presents an integrated system that combines natural language processing with speech understanding in the context of a problem-solving dialogue. It uses a variety of pragmatic knowledge sources to dynamically generate expectations of what a user is likely to say, which includes word recurrence. The way it leverages the recurrence is to transform this context into word expectations that prime the speech recognition system for the next utterance.

Instead of the dialogue strategy, (Young et al., 1989) starts to leverage the phenomenon for recognizing human speech. During the speech recognition, the word recurrence information is used for ASR system to reduce the potential search space for the ASR result. This also expresses a similar idea that identical words tend to occur in close proximity in conversational speech, thus reducing the probability of a specific content word being recognized when it did not occurred. This is similar to the penalization part for our Word Burst rescoring. However this work also does not use any temporal information between different word tokens in their model, their focus is still on whether the word occur in the recent utterances.

2.1.3 Word Recurrence and Further Investigation in Language Modeling

When an ASR system leverages word recurrence, the effects take place in the language model component. Whether a word show up or not in previous context, the probability of the word in language model changes according to the context. As a result, more research efforts are invested in language modeling for leveraging this phenomenon. It also has another name called "Context-Dependent Language Modeling".

(Kupiec, 1989) introduces two complementary models that represent dependencies between words in local and non-local contexts. The non-local context of word dependency considered here is that of word recurrence. The non-local context is used to modify the word transition probability, which is how language model affects the ASR result. (Kuhn and De Mori, 1990) presents a language model that reflects short-term patterns of word use by means of a "cache component", combining with the traditional trigram language model. At this stage, the research efforts start to go beyond identical words, this work also tries to model the relationship between different words that are topically relevant. (Jelinek et al., 1991) describes a simple model based on the trigram frequencies estimated from the partially dictated document that cache the recent history of words, which is similar to the (Kuhn and De Mori, 1990). (Rosenfeld and Huang, 1992) describes a model that attempts to capture within-document word sequence correlations. It used a similar strategy with the earlier cache-based language model, but it extend the coverage of modeling from single word to sequence of words. However, this work also shows that most of the improvement still comes from modeling the exact same word.

The idea that these works are trying to deliver is similar: despite the fact that a standard language model can model the communicated information well, adding more temporal information to the language model can achieve even better performance. Such temporal information includes the recurrence of content words and some extension. The effect of the context will diminish when the communication continues progressing, since the far distance in communication can indicate that stored context might not be relevant anymore. The high-level idea presented here, which is the occurrence of the word that are already observed can affect the probability of other words

around itself, is similar to our approach. But regarding the implementation, the way these works models the temporal information based on the word tokens created by the ASR systems. Our approach models the temporal information based on the elapsed time in conversation, this is one of the core difference. This concludes one of the major difference between conversational speech and formal texts, that there are additional information in conversation such as word recurrence and temporal distance between content words that can be utilized to achieve better understanding.

2.1.4 The Introduction of Word Burstiness

(Church and Gale, 1995) notes that real texts systematically exhibit this phenomenon: a word is more likely to occur again in a document if it has already appeared in the document. Importantly, the burstiness of a word and its semantic content are positively correlated; words that are more informative are also more bursty. (Doyle and Elkan, 2009) leverages this phenomenon to improve topic modeling performance. The phenomenon of burstiness is not limited to text; burstiness also intuitively occurs in other types of data that have been modeled using topic models, including gene expression (Airoldi et al., 2007) and computer vision data (Fei-Fei and Perona, 2005). If a gene is transcribed once in a cell, then it is more likely to be transcribed again. And if a patch with certain properties occurs once in an image, then it is more likely that similar patches will occur again.

Word Burstiness is directly related to our approach in Chapter 4, although there are still fundamental differences between our observation and this line of work. Word Burstiness only relies on a binary condition: whether a specific word has presence in the document or not. It does not include any temporal information in its consideration, as it was identified within documents for Information Retrieval. Our Word Burst originates from our observations in conversational speech, so the temporal information is critical for our approach.

2.1.5 Word Recurrence and Human Perception

Aside from all of the computer system publications we discussed above, there are also research efforts to identify how humans perceive these phenomena in conversations. (Tulving and Schacter, 1990) defines "Priming", which is how humans process these word recurrences. Priming is a non-conscious form of human memory that is concerned with the perceptual identification of words and objects and that has been recognized as separate from other forms of memory or memory systems. The evidence of Priming was showed from experiments including different kinds of tasks, test, type of retrieval cues, kinds of materials and subject populations. Despite it's mostly observed in the experimental environment, this work assumes it occurs in everyday life.

This shows that the phenomenon we discussed in previous sections that had been leveraged in multiple computer systems really exists in human perceptions. The research on human perception validates the usefulness of this idea, and the challenge is how exactly to use it. Our approach in this thesis attempts to allow computer systems to leverage this information by using it as a feature for our tasks.

2.2 Context and Meaning

Our work in Chapter 6 explores the relationship between the words and their surrounding context in conversational speech to identify the meaning of the words. This relationship was presented long ago (Harris, 1954), and there are also multiple published works that leverage this hypothesis.

2.2.1 The Introduction of the Distributional Hypothesis

The Distributional Hypothesis (Harris, 1954) was first presented in 1954. The key concept is that it is possible to define a linguistic structure solely in terms of the "distributions" (= patterns of co-occurrences) of its elements. There is no parallel meaning-structure that can aid in describing formal structure. Meaning is partly a function of distribution.

This is an earlier work that links the distribution of the elements in the data with actual meaning that humans can interpret (Harris, 1954). Following this hypothesis, statistics computed from data can be transformed into meaning that humans can understand. For example, words that occur in the same contexts tend to have similar meanings. There are arguments against this, such as that the words that show up in the same context just have similar usage instead of meaning. (e.q.: red and green tend to occur in a very similar context since they are both colors, yet their meaning is not the same). Still, many research efforts (Bengio et al., 2003; Elman, 1990; Hinton, 1984; Mikolov, 2012) have followed this hypothesis. Our work on Distributed Representations of utterances in this thesis is following this concept.

2.2.2 Distributed Representation of Words

(Hinton, 1984) describes a type of representation in which each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities. Different entities correspond to different patterns of activity over the very same group of computing elements. A partial description activate part of the computing elements, and interactions between the computing elements complete the pattern, thus generating the entity that fit the description the most. A new entity is "stored" by modifying the interactions between the computing elements so as to create a new stable pattern of activity. The main difference between Distributed Representation and conventional computer memory is that the active patterns are not stored anywhere. They can be re-created by using the updated connection strength between different computing elements.

This works turns the word in a large corpus into a point in a high-dimension vector space, and the distance between different points in the space represents the relationships in meaning between different words. This approach can be considered as a way of turning the Distributional Hypothesis into a model that computer systems can use. Our work in Chapter 6 is an extension and application of this representation.

2.2.3 Applications of Distributed Representation

Distributed Representation has become a successful paradigm in multiple applications, including statistical language modeling (Bengio et al., 2003; Elman, 1990; Mikolov, 2012), parsing

(Collobert and Weston, 2008), tagging (Turney et al., 2010), and machine translation (Zou et al., 2013). (Bengio et al., 2003; Mikolov, 2012) all presented high-quality statistical language modeling performance, and each of them use different modeling architectures for creating a Distributed Representation. (Bengio et al., 2003) report on experiments using neural networks for the probability function, showing on two text corpora that the proposed approach significantly improves on state-of-the-art n-gram models, and that the proposed approach allows to take advantage of longer contexts. (Mikolov, 2012) use a simpler skip-gram model for language modeling for Distributed Representation, as it requires much less computing resource yet still achieve high quality result comparing with the other state-of-the-art approach. This model avoids matrix multiplication used in neural network training, and replaced it with a skip-gram model, while both of them models the relationship between the word and its context. These positive results indicate that distributed representation as a strategy for language modeling is really successful, regardless of the detailed modeling architecture. (Collobert and Weston, 2008) presented a single convolutional neural network architecture that, given a sentence, outputs a host of language processing predictions: part-of-speech tags, chunks, named entity tags, semantic roles, semantically similar words, and the likelihood that the sentence makes sense (grammatically and semantically) using a language model. The entire network is trained jointly on all these tasks using weight-sharing, an instance of multitask learning. All the tasks use labeled data except the language model which is learnt from unlabeled text and represents a novel form of semi-supervised learning for the shared tasks. (Turney et al., 2010) presents results on using distributed representations for semantic processing of text. It survey three different types of Distributed Representation, which are based on termdocument, wordcontext, and pairpattern matrices of the documents, for modeling the semantic of a text document. These different modeling approached are applied to multiple applications including document retrieval, document clustering, document classification and lots of other applications. (Collobert et al., 2011) proposes a unified architecture and learning algorithm that can be applied to various natural language processing tasks using distributed representation including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. This versatility is achieved by trying to avoid task-specific engineering and therefore disregarding a lot of prior knowledge. Instead of exploiting man-made input features carefully optimized for each task, their system learns internal representations on the basis of vast amounts of mostly unlabeled training data. (Zou et al., 2013) introduces bilingual word embeddings: semantic embeddings associated across two languages in the context of distributed representation, and they leverage it for phrase-based machine translation. They use a new objective function which embodies both monolingual semantics and bilingual translation equivalence to learn bilingual embeddings.

All of these works demonstrate how distributed representation can benefit different forms of natural language processing, especially from a semantic perspective. We will leverage this representation for our Spoken Word Sense Induction task in Chapter 6.

2.3 Summary

In this chapter, we presented previous work that examined similar ideas to the two human-oriented phenomena that we leverage in the thesis. For the Word Burst phenomenon, since we observed it through conversational speech, we place emphasis on the temporal information,

which is not the focus of previous works. For our work that leverages context to identify the meaning of words, our approach is an application of Distributional Hypothesis to speech tasks. Our work will be focus on spoken language and conversation, which means it will not be applied on formal text, and meaningless word in conversation will be expected. In general, our work will have to be more robust on unclean (either from recognition error or how people talk in conversation) data compare to well-formed text.

Chapter 3

Component Technologies

In this chapter, we discuss the component technologies for this thesis. First, we will introduce Automatic Speech Recognition (ASR). Most of the speech applications use ASR to convert audio into text and then apply different processing. As a result, it can be considered as the foundation for different speech applications. We then discuss the related technique for the Spoken Term Detection (STD) task, which is on of the focus of this thesis. These component technologies show how other people address the task we are working on, and the same task can be approach from different directions. Last but not least, we will discuss the related work for the Spoken Word Sense Induction (SWSI) task. From these component technologies, we can see that the general idea in this task is similar, yet we discover a better model for context information. This chapter maps the progress of the field for the task we are working on.

3.1 Automatic Speech Recognition

Automatic Speech Recognition (ASR) is a fundamental task in the speech processing community. The purpose of ASR is to identify words spoken in the audio stream. Without this transformation, the information contained in the audio stream cannot be easily represented in certain structures, and it is also more difficult to transfer from one person to another (Brown et al., 2001). (Qin, 2013) presents the general framework for an ASR system, as shown in Figure 3.1. The following section references the write up from (Qin, 2013). An ASR system generally 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 outputs decoder output according to the feature representation generated by the front-end. The mathematical formulation introduced in (Jelinek, 1997) are as follows: The ASR system generate the most probable word sequence W from the observed sequence O:

$$W = \arg\max_{W} P(W|O) = \arg\max_{W} \frac{P(W)P(O|W)}{P(O)}$$

$P(W)$ is the prior probability of the word sequence W, $P(O|W)$ is the likelihood of the observation sequence O given the word sequence W, and $P(O)$ is the probability of observing O. Since $P(O)$ is not a variable of W, the above equation can be written as

$$W = \arg\max_{W} P(W)P(O|W)$$

For $P(O|W)$ and $P(W)$, those probability can be estimated from the predefined acoustic and language model, which are the components in the decoder.

The decoder output can be represented in several formats, with each format having its own strengths and weaknesses. Since ASR is not the focus of this thesis, we will briefly introduce each component in a standard ASR system, and the difference between decoder output.

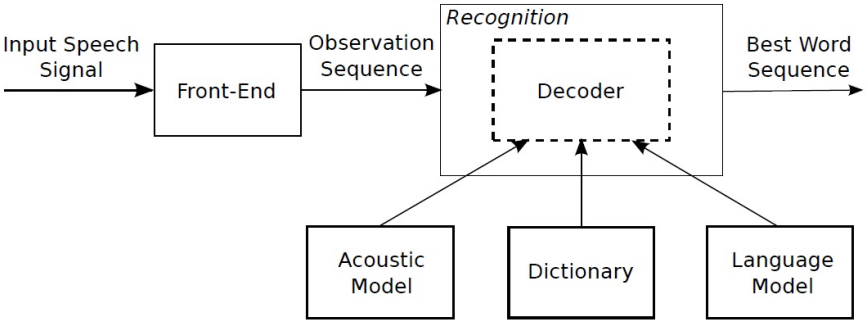

Figure 3.1: General framework for an ASR system (from (Qin, 2013))

3.1.1 Front End

The input speech signal for the front-end is a time-domain sampled speech waveform. This is commonly used for storing speech data. The ASR system tries to simulate how human hearing works, which is based on the characteristics of speech sounds in both frequency and time domain. As a result, a spectral representation of speech signal can be considered to be more appropriate than the time-domain based representation of speech signal for ASR. Since a speech signal is stationary within a short period of time but changes over a longer time (Rabiner and Juang, 1993), we need to segment the input speech signal into small frames when extracting the features. A commonly used frame length is 10 msec, which is considered as long enough to capture the rapid transitions in speech yet has good enough time-domain resolution. The mel-frequency cepstral coefficients (MFCCs) are one of the most popular feature representations in speech recognition, as the mel-scale approximates the human auditory response better. (Davis and Mermelstein, 1980)

3.1.2 Decoder

Following the front-end feature extraction, the decoder computes the most probable word sequence from the extracted feature. A decoder make use of three knowledge sources, which are acoustic model, language model, and dictionary, as shown in Figure 3.1 (Qin, 2013).

Acoustic Model

An acoustic model is used in ASR system to represent the relationship between an audio signal and the phonemes or other linguistic units that make up speech. The model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word.

For the acoustic model, most decoders adopt hidden Markov models (HMMs) (Baum et al., 1967; Baum and Petrie, 1966) to capture the acoustic characteristics of speech data. The HMM parameters can be estimated by using the Baum-Welch (BW) algorithm (Baum et al., 1970), a special case of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

Language Model

The language model is used for obtaining the prior probability of a specific word sequence in a language. In the formula that's presented in section 3.1, it represent the $P(W)$. The most commonly used language model is the *n-gram* language model. It uses a Markov model as an approximation of the true underlying language. It is very helpful to discriminate acoustic ambiguous speech and reduce the search space while decoding. For example, it is very difficult 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 real life.

Dictionary

The dictionary is the third component in a decoder. It is the bridge between the acoustic model and the language model. While the acoustic model and language model work by measuring the different properties of speech in a language, the dictionary links both models with lexical knowledge. The dictionary provides pronunciations of words that maps the relationship between the decoded phones and words, so the decoder knows which HMMs to use for a certain word. It also provides a list of words for the decoder. As a result, an ASR system can only recognize the words present in the dictionary.

3.1.3 Recognition Hypothesis

The decoder generates decoder output based on the input speech data; these hypotheses can be represented in different ways. Aside from the most commonly used lattice, which contains the most information according to the decoder, there are two different decoder output that are also common, single-best recognition hypothesis and confusion network. The single-best recognition hypothesis contains only the most probable recognition result, and its the easiest to be used by other application. The confusion network is a simplified version of lattice, which will be introduced in the following sections. (Chelba et al., 2008) provides an illustration of how lattice and confusion networks represent the same input speech, which are presented in Figures 3.2 and 3.3. The following sections references the write up from (Chen et al., 2013b) and (Mangu et al., 2000).

Lattice

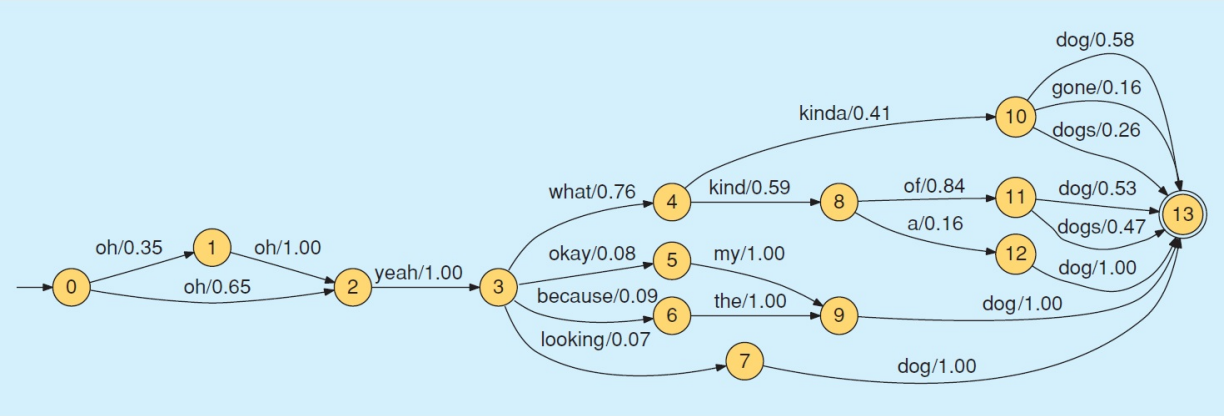

Figure 3.2: An example lattice (from (Chelba et al., 2008))

Lattices are probability networks of the possible decoder output. A lattice contains a set of word hypotheses with boundary times and transitions between different hypotheses (Ortmanns et al., 1997). It can be found that a lattice tends to contain a large number of word hypotheses including both the true hypotheses and the competing hypotheses. The 1-best decoding hypothesis can be created by following the most probable path in the lattice.

Figure 3.2 is an example lattice. As presented in the figure, each edge in the lattice represents a possible recognition hypothesis from a given starting and ending node, while the node in the lattice is the possible word segmentation when selecting a specific path. Between nodes 0 and 2, it is possible that the recognized result is represented as "oh oh" by going through a path of 0 $\rightarrow$ 1 $\rightarrow$ 2 or "oh" by the 0 $\rightarrow$ 2 path. These paths enable the recognition hypothesis to preserve more context information. Another characteristic of lattices can be presented between nodes 10, 11, 12, and 13.There are many edges for the word "dog" between these nodes. This is because their context is different, so the same word could appear in many different places in the lattice. When the same word appears in the decoder output in different time or with different context, the lattice will represent them as different edges.

Confusion networks

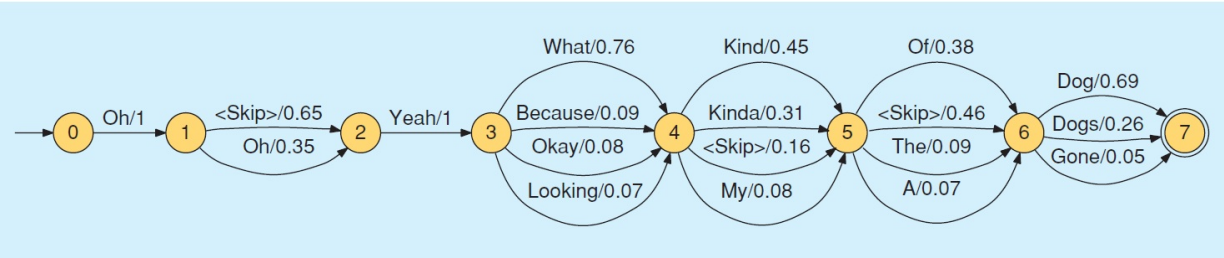

Figure 3.3: An example confusion network (from (Chelba et al., 2008))

Confusion networks (Mangu et al., 1999, 2000) are another recognition hypothesis representation. The motivation behind it is to remove the redundancy in the lattice yet still keep its diversity. It is simpler and more compact compared with a lattice, since it consists of a number of clusters connected sequentially, as shown in Figure 3.3. Each cluster consists of one or more words associated with probabilities. A word in confusion network 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 are then adjusted so that the ending time of each cluster is equal to the starting time of the next cluster.The confusion network is an useful hypothesis representation because it is much easier to search through it, and it contains less duplicated information compare to lattice.

It is possible to convert the lattice into a confusion network, but there are information losses during the conversion, since a lattice contains deeper history information for a recognition hypothesis compared with confusion networks. (Xu et al., 2011) presented a way of creating confusion networks from lattice by using Minimum Bayes Risk decoding algorithm.

3.2 Spoken Term Detection

The experimental setup and evaluation metrics for the STD task were defined in 2006 (Fiscus et al., 2007). Instead of searching the entire document, STD focuses on only detecting the presence of a specific query term. It sounds trivial when people look at it the first time, since a brute force search on the single-best recognition hypothesis seems to handle this problem really well. However, this is not the case when the ASR performance is limited. According to the result presented in (Miller et al., 2007), when the ASR performance is decent (e.g. WER ¡ 15%), searching on the single-best recognition hypothesis also has similar performance as searching on a more complicated recognition hypothesis like lattice and confusion network. However, when a high-quality ASR result is not available, searching into a more complicated recognition hypothesis starts to have significantly better performance than performing brute force search on the single-best hypothesis. As a result, most current STD research follows two different directions. The first involves improving the ASR performance. If the perfect ASR is achievable, STD can be considered as a solved problem. Second, when a high-quality ASR system is not available, most research on STD focuses on searching in more complicated recognition representation like lattice or confusion networks.

Figure 3.4 is the standard STD system architecture and evaluation pipeline presented in (Tejedor et al., 2015). The input speech is first processed with an ASR system, then the recognition hypothesis enters a term searching system to identify the possible instances of our query term. The detection result is then input into the evaluation tool to obtain the evaluation measurement; the most common one is Actual Term Weighted Value (ATWV).

Term searching system in STD

Since we have already identified the need to search through a more complicated recognition hypothesis, we introduce how STD search is done for two of the most common decoder output,

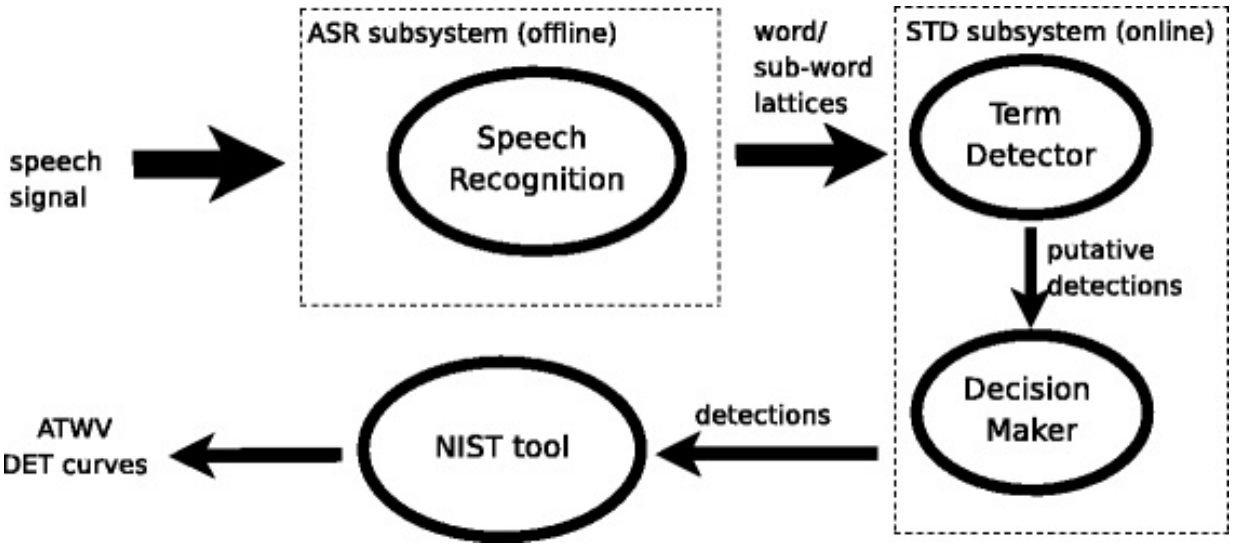

Figure 3.4: Typical STD system architecture and evaluation pipeline (from (Tejedor et al., 2015))

lattice and confusion networks. There are discussions about how deep should we search with the lattice/confusion network; it is a trade-off between the precision and the recall of the STD system. The deeper we search into the recognition hypothesis, the better recall we can expect, yet the precision will also be reduced, as more false alarms will be generated. The ideal search strategy is still an open research question. Regarding the representation of different hypotheses, different retrieval algorithms are applied to it.

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 parts: 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-vocabulary (OOV) queries are mapped to IV queries (proxies) (Chen et al., 2013b) 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 term searching for confusion networks is carried out in another way. For single-word queries, each occurrence of the query word in the confusion networks generates 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, which means no recognition hypothesis is presented. Each path generates detection result: 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 of 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.

Research Progress in STD

The research progress in the STD task can be separated into two stages: Rich Resources Condition (Miller et al., 2007) and Limited Resources Condition (Miao and Metze, 2013). Rich Resources STD is carried out for English, Chinese, and Arabic, which are considered as languages with more resources available. STD systems under this condition do not have a limitation on the amount of training data used for training ASR systems. As a result, a higher-quality recognition result and better STD performance can be expected in the Rich Resources Condition.

In the Limited Resources Condition, the training data for the ASR system are limited to 10 hours, which leads to a relatively high WER (Miao et al., 2013). It is also being tested on languages that do not have huge volumes of linguistic resources available such as Tagalog, Cantonese, Assamese, etc. The bad decoder output heavily affect the performance of the term searching system, since if the query words are not recognized correctly, the term searching system needs to use a special strategy (like phone lattice search) to detect the OOV word. As a result, research efforts (Chiu and Rudnicky, 2013; Chiu et al., 2014; Karakos et al., 2013) are conducted to recover the damage from the Limited Resources Condition to make it as good as the Rich Resources Condition.

There are two major approaches for the STD task, the query-by-example approach (Jansen et al., 2010) and the ASR and term searching two stages approach (Mamou et al., 2013; Miller et al., 2007).The two-stage approach is the current mainstream, yet we introduce both of them in the following section after the task condition introduction.

3.2.1 Rich Resources Condition

The STD research in the Rich Resources Condition establishes the standard pipeline for STD systems, since the ASR and term searching two stages approach that comes from SDR task has overall better performance compared with the query-by-example (Wang et al., 2013) approach. (Miller et al., 2007) presents an STD system that searches 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 recognition hypothesis (James and Young, 1994). (Mamou et al., 2007) presents a vocabulary-independent system that can process arbitrary queries, exploiting the information provided by having both word recognition hypothesis and phonetic recognition hypothesis. This system is based on word confusion networks and phonetic lattices. (Vergyri et al., 2007) reported their system for the STD 2006 task; they analyze the effectiveness of different index ranking schemes, and the utility of approaches to deal with OOV terms.

Despite those systems being different in detail, the overall structures are similar. The ASR system and the term searching system are two indispensable components for a successful STD system. Lattices and confusion networks have started being used as the better recognition hypothesis instead of the one-best hypotheses, due to the rich information contained in them. Different search strategies are applied to the OOV queries to compensate the inability of ASR systems to

recognize OOV words. The most common approach is to perform search on a phonetic recognition hypothesis instead of a word recognition hypothesis (Mamou et al., 2013; Miller et al., 2007). These discoveries are still valid, as the state of the art STD systems still use a similar pipeline and strategy.

The STD performance in the Rich Resources Condition can be good. The best system can achieve around 80% of the maximum possible accuracy score in English. However, good results demand high-quality ASR output. What if high-quality ASR results are not available? This question leads STD research into the next stage: the Limited Resources Condition.

3.2.2 Limited Resources Condition

STD under the Limited Resources Condition has been proposed as a research challenge recently, and a focus in the current speech community (Karakos et al., 2013; Mamou et al., 2013; Miao et al., 2013). The ASR system with limited training data is unable to generate a high-quality decoding result (e.g. when there is only 10 hours of training data available.) Hence, a low-quality hypothesis limits the achievable performance for the term searching system, and ends up having far worse STD performance compared with STD under the Rich Resources Condition. The following are a few approaches people had proposed recently to improve STD under the Limited Resources Condition.

Deep Neural Network-based Decoder

Deep Neural Networks (DNN) have been widely used in ASR recently as a better acoustic modeling technique. (Dahl et al., 2012) first proposed a novel context-dependent (CD) model for ASR. It introduced 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. (Miao and Metze, 2013; Miao et al., 2013; Zhang et al., 2014) further extend the usage of DNN with dropout and maxout techniques to make it further robust under the Limited Resources Condition. Another approach is to include multilingual information using the DNN from a Rich Resources language to improve a Limited Resources language's performance (Knill et al., 2013), it can be done by training the deep learning model on language with richer resources, then applying the model on the Limited Resource language. All of the related research provides solid improvement for ASR under the Limited Resources Condition, yet the WER is still very high, as the Limited Resources Condition severely degrades the decoding quality. As a result, this also inspires us to work on the other part of the STD problem, as so many research efforts have already been invested in ASR, yet the improvement is still limited.

Hypothesis Rescoring

Aside from improving ASR, one way to improve STD performance focuses on rescoring the decoder output according to different features or information. (Mangu and Padmanabhan, 2001) used transformation-based learning and lexical features to improve WER from the two best hypotheses in a CN confusion bin. Transformation-based learning requires a set of allowable trans-

formation type and an objective function to pick the most idea transformation for the given data. Similarly, (Allauzen, 2007) detects errors on broadcast news transcriptions using lexical, syntactic, and contextual information. (Tur et al., 2013) trained conditional random fields using CNs instead of the 1-best transcription to improve accuracy in slot-filling in semantic frames. (Stoyanchev et al., 2012) used syntactic and prosodic features to identify mis-recognized words to generate clarification questions in speech-to-speech translation. (Chen et al., 2011) proposed graph-based re-ranking for STD by using acoustic similarity. (Mamou et al., 2013) proposed a hypotheses rescoring algorithm based on the other retrieval results from the same query. The concepts for most of these efforts are the same: Since the quality of hypotheses is bad, applying other knowledge or information sources is beneficial for fixing the errors created by the bad ASR system. Our work in Chapter 4 follows this line of thought, as we use our knowledge of the Word Burst phenomenon in conversational speech as a source of information to perform recognition hypothesis rescoring.

System Combination

The other way to improve STD performance without enhancing the ASR system focuses on combining the ASR results from multiple systems to achieve a better STD result. The reason why this approach can achieve better result is because different ASR systems usually have different parameters, and each usually has it's own unique correct/error hypothesis. The diversity from different systems can be integrated together to create a better result compared with each of the individual systems. (Mangu et al., 2013) produces complementary STD systems and shows that the performance of the combined system is 3 times better than the best individual system. (Mamou et al., 2013) investigates the problem of extending data fusion methodologies from Information Retrieval for Spoken Term Detection on Limited Resources Condition. (Karakos et al., 2013) performs system combination, where the detections of multiple systems are merged together, and their scores are interpolated with weights that are optimized using the evaluation metrics. The combination technique combines the score for each detection to generate a new detection list that combines the detection score from each individual system. The research shows that the integration of diverse systems can contribute to better overall performance in STD, because the combined result has better recall for the target, yet still has a similar level of precision. Our work in Chapter 5 follows this idea, as we use the different decoder output as a source of new ways of combination from the same ASR system.

Morphology-based OOV detection

The goal for OOV detection is trying to break a word into a smaller unit, and performs the search based on the smaller unit. Morphology has started to catch the attention of the STD research community recently. The research on morphology in limited resource STD is mostly focused on Turkish and Zulu. (Narasimhan et al., 2014) proposed using morphemes in STD for OOV queries. The most common solution for OOV queries is to use phonetic search (Mamou et al., 2007; Miller et al., 2007) instead of the regular word-based ASR result. However, the performance is limited, because the phonetic unit is too small. This approach adds morphemes into the vocabulary and performs a hybrid decoding. During the search, it converts the OOV keyword

into a sequence of morphemes, and performs morpheme-based search on the hybrid decoding result. It shows significant improvement on the Turkish OOV keyword search performance. Our work in Chapter 4 also achieves better performance on Turkish, yet our improvement is on the IV word, since our system does not process OOV words at all. (Chen et al., 2013a) proposed morphology-driven lexicon expansion for STD systems. This approach models the productivity of morphological phenomena, and is combined with a G2P system for OOV word reconstruction. The second pass decoding with reconstructed words shows better performance on STD in Zulu. In general, most of these works that leverage sub-word units like phones or morphemes can achieve improvements on the OOV keyword search, as the OOV keyword search is not covered in the standard ASR and term searching pipeline. Yet for the IV keyword, these approaches usually will not be very helpful. Still, this can be considered as a reasonable solution specifically for OOV words.

3.2.3 Query-by-example STD

The query-by-example approach focuses on pattern matching of spoken term queries at the acoustic level. The motivation behind this line of research is as follows: If we can perform signal-level matching between the speech template we already have and our target corpus, then we can perform STD without any external resource, or the so-called "Zero Resource" condition (Jansen et al., 2013) will be possible. The research (Hazen et al., 2009; Jansen et al., 2010; Zhang and Glass, 2009) in this direction focuses on the condition where limited or no in-domain training material is available and accurate ASR is unavailable. Query and the target corpus can be represented in different parameterizations of the speech templates such as the raw MFCC features or phonetic posteriorgrams generated by different phonetic recognizers. Query matches in the target corpus are located using a modified dynamic time warping search between query templates and target corpus.

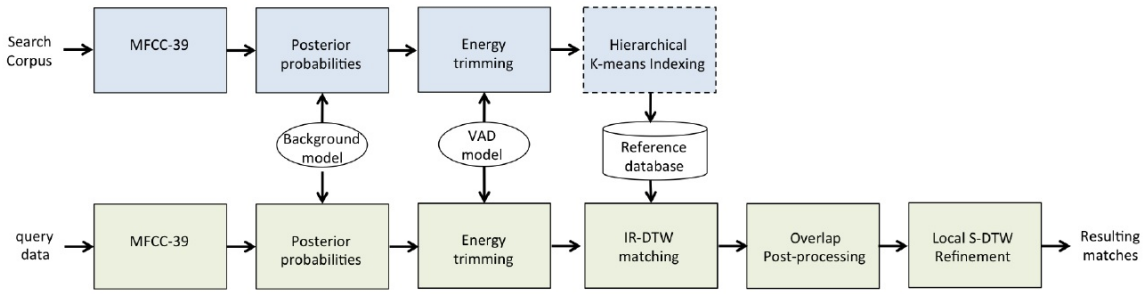

Figure 3.5: An example system framework for a query-by-example STD system (from (Anguera et al., 2013))

Figure 3.5 is an example of a query-by-example STD system's framework that is presented in (Anguera et al., 2013). The system performs feature extraction on both searching corpus and spoken query. Post-processing can be applied to the extracted feature to make it more robust. In this example system, it uses energy-based Voice Activity Detection (VAD) to trim off the

silence/non-speech portion of the speech data. The processed data are then indexed and searched with the DTW approach. When multiple instances of spoken query are available, the DTW search will be performed on all different instances. After the search on multiple instances, the system will merge the result and only output the one with the highest score.

However, there are several well-known issues in this approach, and those issues make this approach difficult to compete with the current state-of-the-art ASR and term searching approach. First, the robustness is not as good as the other ASR and term searching based approach. The search completely depends on the acoustic template for the query. It can achieve decent performance on the data recorded by the same speaker and in a similar environment. However, the acoustics of the same word can be very different for many reasons: speakers, recording device/environment, the length of the word, etc. These acoustic variants limit the generality of the searched result, since each query template only detects the part that sounds exactly like itself. Second, the computation is very expensive. This approach requires comparing each frame of the speech to identify the distance of two speech templates. When processing with multiple templates on a huge speech corpus, the computation requirement increases rapidly. Last but not least, in order to perform STD with this approach, a speech template as query is needed. However, given the condition on which it claimed to focus, where limited or no in-domain training material is available, it is difficult to obtain the required query speech template, especially when you are searching on lots of different queries. As a result, this approach is mostly applied to the situation where less query template are required.

Recently, (Lee et al., 2014) proposed a graph-based re-ranking approach based on the concept that search results, which are acoustically similar to other results with higher confidence scores, should have higher scores themselves. This follows the query-by-example idea, but uses the acoustic information for rescoring the result coming from an ASR and term searching based STD system. This avoids many issues that occur in the standard query-by-example STD setup, as the acoustic distance is only computed on a small portion of data that is identified by the ASR and term searching based system; computation is no longer a significant issue. Also, the ASR and term searching based system will identify the possible location for the query and extract those as templates, so that template generation is also done automatically. This approach provides improvement on the STD performance for OOV words, since the ASR system cannot process it at the word level, and phonetic transcription is more noisy. However, when the query is an IV word, it cannot provide statistically significant improvement on the performance.

In general, the query-by-example based approach can provide good improvement on STD when the standard ASR system cannot address that condition too well, such as on OOV words. However, when the query can be processed within the standard pipeline, the query-by-example approach becomes less effective. (Lee et al., 2014)

3.3 Spoken Word Sense Induction

SWSI is the task of automatically identifying the senses of spoken words without the need for handcrafted resources or manually annotated data. A SWSI system usually includes two major components: the ASR system and the Word Sense Induction (WSI) system. We already introduced the ASR system earlier this chapter.In this section, we focus on the WSI system which is

similar to a standard document clustering system.

Before discussing the related work for WSI, we first introduce Word Sense Disambiguation (WSD), which can be considered as a precursor to the WSI task.

3.3.1 WSD

WSD is introduced in section 1.3.3. The result obtained from this task can impact other computer-related writing, such as discourse and improving the relevance of search engines. The general strategy addressing WSD is shown in in Figure 3.6[1]. The figure demonstrates how a WSD system disambiguate multiple instances of the word "kiwi" as the fruit or the bird that cannot fly in Australia.

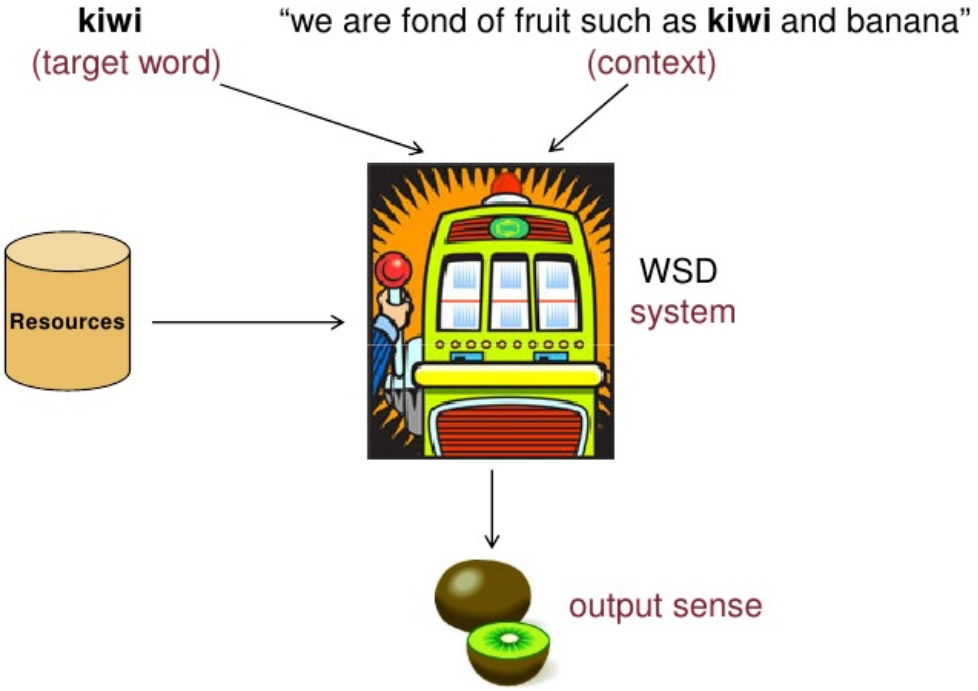

Figure 3.6: Illustrative example of the general framework for WSD task (from footnote 1)

We input the target word and its context into a WSD system, and the system leverages its (usually human-labeled) resource to identify the sense of the given target word according to the context. Since most of the approaches here require sense-tagged data provided by humans, porting WSD systems from one language to another requires significant effort to collect the required data. The following are a few approaches that have been presented to address the WSD task.

[1] http://naviglinlp.blogspot.com/2012/05/lecture-14-project-presentation-and.html

Supervised approach

Supervised WSD is similar to a standard Machine Learning classification task. It requires labeled training data to train a classifier for each target word. Support Vector Machines (SVMs) is one of the most successful approaches so far (Zhong and Ng, 2010), because it can cope with the high dimensionality of feature space properly. However, due to the fact that it cannot adapt to new languages without retraining the classifier (which requires annotated data from the new language), its generality is limited. (Khapra et al., 2009) tried to use a trained model from one language to test on another language, and the WSD performance was poor, which demonstrated the limitation on generality for the supervised WSD approach. These approaches can be considered as very standard Machine Learning applications, and good results can be achieved when the appropriate training data are available. However, obtaining appropriate data is never an easy task.

Knowledge-based approach

The knowledge-based approach uses existing knowledge resources with specific human-designed rules for WSD. (Navigli and Lapata, 2010) proposed using Degree on WordNet (Fellbaum, 1998) and BabelNet (Navigli and Ponzetto, 2010) to create a semantic graph and use the structure of the graph for WSD. This approach has a similar limitation with the Supervised approach, which is difficult to adapt to new languages due to the lack of existing data. Still, it is capable of achieving good performance on an English WSD task (Navigli and Vannella, 2013). The lesson we learned from these approaches is similar to what we learned in the supervised approaches: when the appropriate training data are available, good WSD performance can be achieved in automatic systems, yet the labeled training data might not always be available.

Semi-Supervised approach

(Yarowsky, 1995) presents the most famous Semi-Supervised method that uses a limited sense inventory, a few sense-labeled examples, and an unlabeled corpus to create a WSD system. The system uses the known example to predict the word sense for the word in the unlabeled corpus, similar to the concept of relevance-feedback (Rocchio, 1971). This is one of the earliest works to leverage the unannotated data to disambiguate the meaning of a keyword. It shows that with a very limited amount of sense-tagged data, we can still achieve decent performance on the WSD task. This can be considered as one of the pioneers for the WSI task, since removing the small initial sense-tagged examples can make this approach meet the requirement for a WSI approach, which does not use any human labeled data.

3.3.2 WSI

A WSI system can be considered to be an unsupervised WSD technique using Machine Learning methods on raw data without relying on any external resources such as a dictionary or sense-tagged data. The algorithm usually infers word sense from data by clustering keyword instances following the Distributional Hypothesis (Harris, 1954), which is popularized with the phrase "a word is characterized by the company it keeps" (Firth, 1968). Figure 3.7 (Liu et al., 2012) shows

a general framework for a text clustering system, and for SWSI, the text just needs to be replaced with the output generated by an ASR system.

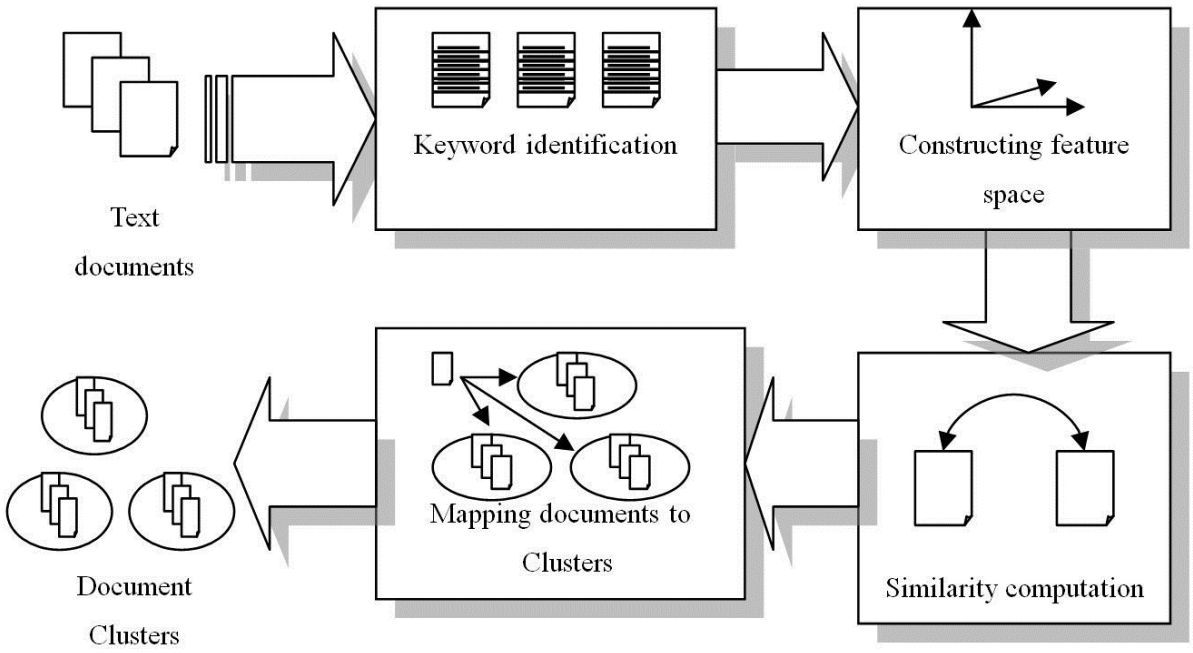

Figure 3.7: A general framework for text clustering/WSI system (from (Liu et al., 2012))

The "Constructing feature space" block and the "Similarity comparision" blocks in the figure are two main steps for document clustering. The keyword identification step cleans up the raw input data, including stop word removal and stemming. The feature space for clustering is then extracted from the cleaned-up data, and the clustering is performed based on the extracted feature. Most WSI systems extract the different senses of words following one of these two approaches:

- Local approach: Cluster the instances of a keyword solely according to the context that co-occurs with the keyword
- Global approach: Represent the instance of a keyword and its context according to a model trained with a larger corpus. This approach usually requires more data but can achieve better performance.

It is worth noting that including more unlabeled data into a WSI system can usually achieve better performance, as there are more data available for learning. Since the distinction between WSD and WSI involves whether the labeled resource such as a dictionary or sense-tagged data are used, adding unlabeled data does not violate the definition of WSI. Still, in this thesis, we do not focus on adding extra unlabeled data to our SWSI system, as our goal is to discover a better way of representing the data. Most of the works in the field can be separated into a few categories according to the clustering strategies.

Simple Clustering approach

The assumption for this line of research is that words are semantically similar if they appear in similar contexts. (Lin, 1998) used syntactic dependency statistics between words that occur in a corpus to produce a set for each discovered sense of a target word. By defining a similarity function, several clustering algorithms are applied to word feature vectors (Pantel and Lin, 2002), such as K-means, Bisecting K-means (Steinbach et al., 2000), Average-link, Buckshot, and UNICON (Lin and Pantel, 2001). Clustering by committee (Pantel and Lin, 2002) also uses syntactic contexts for the task of sense induction, but uses a similarity matrix to encode the similarities between words. It relies on the notion of committees to output the different senses of the word of interest. These approaches use simple methods to represent the context for each word and standard clustering technique for the WSI task. The assumption still holds, yet there is more research to be done on more complicated features or customized clustering techniques for the WSI task.

Extended Clustering approach

This line of research considers that words tend to manifest one sense per collocation (Yarowsky, 1995). A good example of the extended clustering approach is the context-group discrimination algorithm (Schütze, 1998) that is based on large matrix computation methods. (Pinto et al., 2007) tried to improve the utility of small corpora through self-term expansion. (Brody and Lapata, 2009) frames the WSI task in a Bayesian context by considering contexts of ambiguous words to be samples from a multinomial distribution. More recently, (Pedersen, 2013) reported their WSI systems based on second order co-occurrence features. Generally speaking, these approaches can be considered as an improvement on features or clustering algorithms for a simple clustering approach. This is still one of the most popular approaches even for recent WSI systems, as the intuition of it is clear and easy to understand. Our work in the thesis also belongs to this category, as we aim to obtain a more robust feature for the SWSI task.

Graph-based approach

Most of the graph-based approaches assume that the semantics of a word are represented by a co-occurrence graph, where nodes are co-occurrences and edges are co-occurrence relations. The co-occurrences between words can be captured through the basis of grammatical relations (In this paper, means different Part-of-Speech tagging pairs) (Widdows and Dorow, 2002) or collocation relations (Véronis, 2004). HyperLex (Véronis, 2004) is considered as a good graph-based algorithm, based on the identification of hubs in co-occurrence graphs that have to cope with the need to tune a large number of parameters (Agirre et al., 2006). There are also multiple clustering algorithms being proposed for the graph-based model, including Curvature Clustering (Dorow et al., 2004); Squares, Triangle, and Diamonds (SquaT++) (Navigli and Crisafulli, 2010); and Balanced Maximum Spanning Tree Clustering (B-MST) (Di Marco and Navigli, 2013). These aim at identifying word meaning using the local structural properties of the co-occurrence graph. Although concurrence can be considered as a way of representing the context, we separate this into a different sub-section other than an extended clustering approach because there are many research efforts invested in this phenomenon.

Graphical Model / Optimization approach

Ever since its introduction, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) has been applied to many different language processing tasks. (Lau et al., 2013) presents a WSI system based on LDA modeling. In order to decide the number of topics T, which is originally a parameter in LDA, it relaxes this assumption by extending the model to be non-parametric, using a Hierarchical Dirichlet Process (Teh et al., 2006). These approaches achieved very good performance in a recent WSI shared task (Navigli and Vannella, 2013). On the other hand, (Pedersen, 2013) treated the WSI task as a submodular function maximization problem, which also achieved good performance on WSI recently. These works model the context of target words with different models, yet the general idea is still the same, using the context to identify different meaning. They demonstrate that graphical models can be used as a better feature for clustering. These are also part of the baseline system we present in our thesis.

Translation-based approach

Every WSI approach we described above requires monolingual data. When multilingual data are available, research efforts have been undertaken to incorporate multilingual data for the WSI task. Translation-based WSI involves augmenting the source language context with target language equivalents. This approach assume the semantic information in the parallel corpus are identical. It aims to capture more semantic info comparing to a single language by bridging the parallel corpus with Machine Translation approach. (Apidianaki, 2008) uses a bilingual parallel corpus to construct two dictionaries, where each word type is associated with its translation equivalents. The lexicon is filtered in such a way that words and their translation equivalents have matching POS tags and words appear in the translation lexicons for both dictionaries. The result outperforms the baseline system presented in the paper, yet the paper only report the results on five keywords, which could be insufficient. Still, we anticipate that the effectiveness of these categories of research will be limited, as a parallel corpus is not easy to obtain.

The comparison between our approach and these related works will be discussed in section 6.3.3.

3.4 Summary

In this chapter, we reviewed the task-level related literature for the thesis, including Automatic Speech Recognition (ASR), Spoken Term Detection (STD), and Spoken Word Sense Induction (SWSI). For the speech-oriented tasks we have introduced, the state of the field is already able to produce a robust system for clean data if a substantial amount of training data is available, since simple approaches that were explored earlier such as the n-gram language model can perform very well under easier conditions. When processing on noisier data or when fewer training data are available, the performance starts to drop as these modeling approaches are unable to perform well under more difficult conditions. This is relevant to the goal for our thesis, which is trying to identify a robust and easy-to-adapt method to process these noisy speech data. Some research starts to introduce external knowledge such as context words or topics to improve the performance under difficult conditions. This is commonly seen in the literature we discussed

in this chapter. These kind of approaches base on a similar idea, that the performance for the task could be improved if extra information are included in the model/algorithm. The context or topic information can be extracted from a bigger corpus, which can let the knowledge benefiting from larger amount of data. In our thesis, the external knowledge we introduce are the different phenomena that we can discover in human conversation or automatic systems, as those do not really require external training data to use.

Chapter 4

Conversational Word Burst

4.1 Motivation

Recent advances in Spoken Term Detection (STD) focus on the Automatic Speech Recognition (ASR) system. By improving the quality of ASR result, we can also achieve better performance on STD task. However, the Word Error Rate (WER) for the ASR system is still relatively high in several situations. One such condition is Limited Resources Condition, which limits the quantity of training data for the ASR system. In Limited Resources Condition, limited amount (less than 10 hours) of training data are available, hence it is more challenging to create robust model from it. This condition can also be observed in real world application when we are trying to search the content in a less common language that have sufficient amount of data. It is also very difficult to continue improving the ASR performance, as it is been explored by many researchers, with diminishing returns. As a result, this thesis approaches the problem from a different perspective. Once a decoding is produced, can we introduce external information to help us refine the noisy recognition hypothesis? (Chiu and Rudnicky, 2013) The information we leverage also needs to be language-independent, so that it can be apply to multiple languages. We think that Word Burst phenomenon is one of the forms of external information that match this need, as it is how humans organize their conversational speech. Word Burst describes the phenomenon in which a word that has been spoken recently in conversation is more likely to recur in close proximity. The structure of speech utterance should be consistent regardless of language; we can use this information to identify whether part of the delivered information is not useful, and focus our system more on the part that matches conversational structure. The reason why we believe it can be an useful feature is because conversation typically have topics, and people will talk about relevant topic for a period of time.

4.2 Our approach

Figure 4.1 shows where our approach fit into a standard STD pipeline. The ASR system output lattices/confusion networks serve as decoder output for the incoming term detection/search step. Our goal here is to leverage the Word Burst phenomenon to improve the quality of the recognition hypothesis, aiming to achieve better STD performance. We rescore word hypotheses according

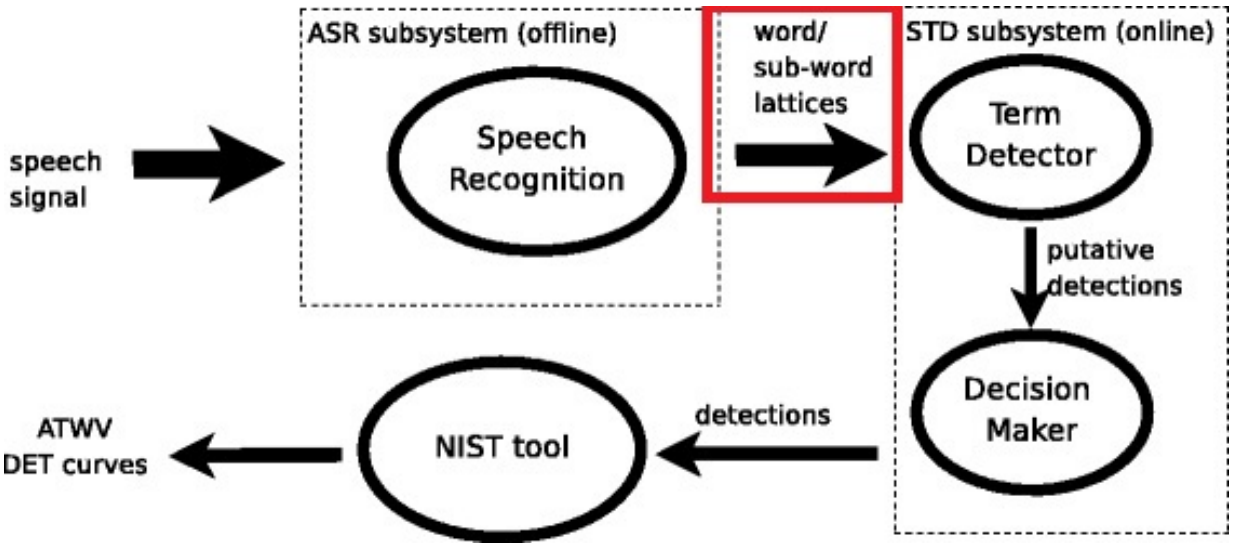

Figure 4.1: The components in the standard STD pipeline used in the BABEL program (Karakos et al., 2013; Mamou et al., 2013). See Chapter 3 for a discussion of the individual processing steps.

to the Word Burst phenomenon. This thesis introduces two assumptions that flow from the proprieties of the Word Burst phenomenon:

- The information that is delivered in close proximity tends to have elements that are relevant to each other, so the same word is likely to cluster up within a small temporal window. This is the key conversational feature in this thesis.
- A word that appears alone without other identical instances in close proximity tends to be noise caused by recognition errors

These two assumptions are actually the flip side of the same coin. If the same word shows up multiple times within a small time window, it means that all of these instances shares similar context and hence they are all relevant, so these words could all be true. On the other hand, if a word shows up by itself without any other identical word, this means the context may not seem to be relevant for that word (so that there is no other instance of the same unit here), then it is possibly a recognition error. We understand that these are very strong claims, and whether these can contribute to our system also strongly depends on the data we employ. Still, the idea behind these assumptions is that if we can anticipate the pattern of a word's occurrences, then we can use that knowledge to determine whether a machine-recognized output of a human conversation violates these assumptions. If it does, then there could be recognition errors present, because we do not expect people to typically talk in that way.

Given our assumption is based on time window, a common question that will show up is, why we are using time window instead of token distance to model the relationship between a word and its content? There are a few reasons for us to make this decision. (Bengio et al., 2003) describes it uses word tokens to capture temporal structure between different words. Our approach directly uses temporal information, which is more precisely on modeling the temporal information. Moreover, our approach will be applied on to different languages, and the power

of word tokens can varies between different languages. In an agglutinative language, a single token can represent rich meaning that can only be described by multiple word tokens in other languages. This makes word token being less ideal of a unit to capture the relationship between a word and its context if we want to create an algorithm that works on different languages. (Banerjee and Rudnicky, 2004) described "The tree learnt at the 20 second mark also revealed the number of speaker changes to be the topmost node, implying that that is one of the most important features for detecting the state of the meeting.", indicating temporal window is a good feature to understand the state of meeting. As a result, our algorithm uses time window instead of token distances.

4.3 Word Burst and Unique Penalization Rescoring

We designed two different recognition hypothesis rescoring algorithms, the Word Burst Rescoring and the Unique Penalization, to leverage the Word Burst phenomenon. The two assumptions: "The information that is delivered in close proximity tends to be relevant to each other, so the same word is likely to cluster up within a small temporal window" and "A word that appears alone without other identical instances in close proximity is more likely to be noise" are implemented in these rescoring algorithms. Word Burst rescoring focuses on identifying a good balance between the two assumptions by increasing the hypothesis score of the word when it has nearby identical instance and penalize the hypothesis score of the word when it doesn't. Unique Penalization rescoring focuses on trying to identify the most ideal context size for the penalization part of the assumption. We select different context size to apply the penalty according to whether a recognition hypothesis appears without other identical instance.

4.3.1 Word Burst Rescoring

We observe that conversations tend to focus on particular topics; the high likelihood of a content word related to the current topic occurring near other instances of the same word is called Word Burst (Chiu and Rudnicky, 2013). More precisely, when in a conversation that touches on specific topics, the content words within the same topic will tend to occur near each other. The reason we emphasize content words is because the function words in spoken language occur too frequently, and Word Burst cannot provide too much extra information because it already shows up everywhere. A similar phenomenon was described by (Church and Gale, 1995); however they focused on text materials, specifically under the Information Retrieval setup, which large amount of well-formed documents are available. The current thesis explores this phenomenon in the context of spontaneous conversations.

We define a "content" word in terms of frequency; that is, the most frequent words in an available corpus are designated as "function" words. 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 a word not in the stop list is considered as a content word. The benefit of this approach is that since we only have a very limited amount of vocabulary available (due to the limited amount of training data we have) in the Limited Resource Condition, it is not possible to identify all of the content words.

However, since the function word or stop word will always show up frequently in any amount of data because of the way humans speaks, identifying the common words that show up in any corpus can give us a good idea about which words are less likely to be the content word. The detail of deciding the size of stopword list will be discussed in the later Algorithm Development section. The experiments are reported at section 4.6.1.

Evidence of Word Burst

The assumption in the Word Burst phenomenon is only an assumption if there is no actual evidence to support it. Hence, we examined our data and showed several forms of evidence that suggest that Word Burst exists in conversational speech.

Table 4.1: Content word window size / burst percentage.

	10 sec	**15 sec**	**20 sec**	**25 sec**	**30 sec**
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

(Banerjee and Rudnicky, 2004) proposed using a window size of 20 seconds 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 words that have another instance of the same word that appears within 20 seconds. We exclude words from the top 1% and all singletons in the available corpus and consider the rest of the word as content words. As can be observed, content words (as defined) tend to occur in bursts. Among the different languages we analysis, Turkish has low percentage of content word, since it is an agglutinative language, and word tends to recur in morphological variants, hence it is harder to have Word Burst on Turkish.

Another form of evidence is to look at the distribution of a specific word in our data. Figure 4.2 provides a visual example, using the distribution of the Tagalog term *magkano* in our data. In the graph, we can see that the words have the tendency to occur in bursts. However, this needs to vary according to language. In an agglutinative language such as Turkish, words can appear as morphological variants and thus require a longer stop-word list or a better way of normalizing the word back into the simple form, instead of only performing exact word matching.

The reason why we did not leverage topical word information to identify Word Burst is that, since we are processing data in the Limited Resources languages, we do not know the meaning of the word token in those languages. Even if the topical word can be identified by a trained topical models, it is difficult to train a high quality topical model with limited amount of training data available (Which is the condition we are working with in this chapter.)

Algorithm Development Process

After we established that the Word Burst phenomenon exists, the next question is how to develop an algorithm to leverage this phenomenon to improve the quality of recognition hypotheses.

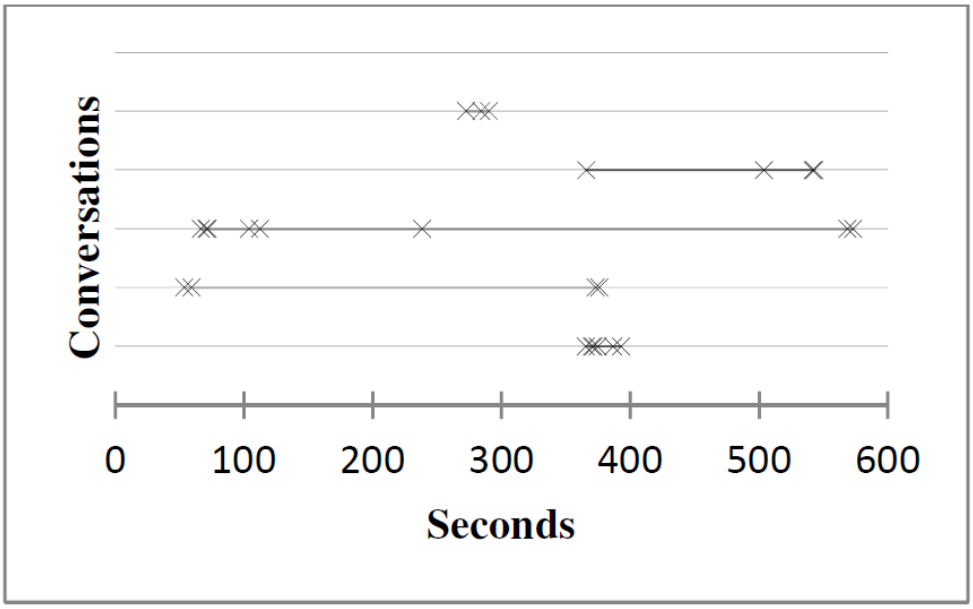

Figure 4.2: Term incidence for Tagalog *magkano*, which means "How much?" in English (x-axis is time of conversation in seconds; each individual line represents separate conversations, and the crosses on the lines are the locations where an instance of *magkano* occurs)

First, we believe leveraging context is very important, especially identical words that occurs in the neighbors. That is how Word Burst phenomenon expect the distribution of the content word. Therefore, every recognition hypothesis received a bonus if there are other instance of the same word nearby within a predefined window size. If not, it receives penalty. Our proposed algorithm is similar to how the Word Burstiness are leveraged in (Church and Gale, 1995), yet we limit the effect of coocurrence bonus from full document to a localized context. This initial version of the rescoring reduces our system's performance, since after the rescoring the recognition hypothesis score of stopwords usually received lots of bonus due to the high frequency it shows up in the conversation. Our analysis in 4.3.1 defines Word Burst only focus on the content word, since stop word usually occurs too frequent and is less informative for understanding the conversational topics. As a result, a stop-word list is used to minimize the over-bonus problem. With this setup, we are able to achieve positive result, which is reported in (Chiu and Rudnicky, 2013). In addition, every rescoring algorithm is based on an existing system, it is also important to find a balance between trusting the original system and rescoring to achieve better performance. Receiving bonus from the recognition errors might cause false alarm being spread to other recognition hypothesis, and there are always words that just tend to show up alone, penalizing every word that does not have a instance of itself in neighbor will cause the overkill of the recognition hypothesis score. As a result, we also introduce a threshold to determine whether or not to use the rescoring algorithm or not. The threshold is computed by tuning on the development data.

Deciding how to apply the Word Burst phenomenon in our rescoring algorithm is also challenging. When having many different input features, computing a weighted sum as output between different input features is a common approach, which is similar to the computation used in

Deep Neural Networks (Hinton et al., 2012). The first attempt was to add or subtract a fixed probability when these condition were met. The problem coming out of this approach is that, since the score of recognition hypothesis is a probability, adding / subtracting a fixed value could let the score go above 1 or below 0, which violate the property of probability. Of course this can be fixed by bound the score to 0 and 1 if it goes beyond the bound after rescoring, but this approach also has another limitation: it assume all the bonus and penalization are identical, and the only thing matter is the number of occurrence of the other identical words. This is relatively far from the concept of weighted sum of the input features, as the computation of the output result is not affected by the property of other input contexts. This can be fixed by deciding the bonus/penalty according to the property of context (other instance of the same word around rescoring target).

To include more information from the input features to get a better weighted sum for the output, we investigated the features that can be used for rescoring, and we identify three features that could be leveraged for our algorithm: the distance between our rescoring target and the other instance, the probability of our context that will be leveraged for rescoring and the number of instance around our rescoring target. The intuition behind these factors are described as follows: The distance between our rescoring target and its context represent how the topical information fades according to time, and the bonus should reach zero when it is at the border of our window size. We did not find a function that's better than linear on both performance and explanation, so a linear function is been used as the function to represent the relationship on distance between rescoring target and it's context. The probability of the context indicates our confidence about the context should also affect how much bonus the rescoring target can receive from its context. Having a word that's very likely there should gain more bonus comparing to having a word that has much lower recognition probability around, since the latter is more likely to be a recognition error.

A weighted sum is computed by multiplication on the input feature then summing up as the output. To simulate this operation, we consider the score of the original hypothesis is always given a weight of 1, and its context provided bonus based on its features that are described in the last paragraph. To enhance the importance of multiple instance of identical words showing up in near context, the score provided by the context is further multiplied with an extra factor, which represents the fact that, if there are more instances of the same word occurring in a small region, it is a even stronger indication that the word really exists there, so these extra instances should receive a further bonus. With tuning on the development data, we obtained the power of e as this extra bonus function, as the exponential grow represent the extra bonus for the same word showing up in close proximity. When it's lacking identical words in the context, we reduce the weight for the original hypothesis, since it is violating the Word Burst assumptions.

After the rescoring, there's a normalization step to normalize the probability of all hypotheses that shows up in the same time. The normalization compute the sum of the rescored hypothesis score and divide individual score to the sum to make sure the recognition hypothesis score is still a probability, and being compatible the the search system we use. This is also useful for avoiding score going above 1, which is commonly seen when there are several identical word clustered together, as the weighted sum have no upper bound of the possible score. Through normalization we can still maintain the relative relationship between different hypothesis yet bound the score back to the probability space.

The parameter tuning process is the key to make Word Burst rescoring algorithm being ef-

fective. Due to there being multiple parameters (window size, threshold, penalization or bonus parameter) that need to be tuned, we use grid search on these parameters and evaluate the performance on development data. While doing grid search, we only move one parameter with a small constant while keeping other parameter fixed and see which value provide the best performance on development data. We iterate through each parameter in this way to obtain a local optimal parameter sets.

Rescoring Algorithm

Word Burst rescoring relies on our two assumptions: The same word is likely to cluster up within a small temporal window, and a word that appears alone without other identical instances in close proximity tends to be noise. When converting this assumption into the recognition hypothesis rescoring algorithm, the algorithm manipulates the hypothesis score according to the presence of the other identical words. For each hypothesis, if there is another hypothesis for the same word temporally close, it is assigned a score bonus; if there is no other hypothesis of the same word close to it, it receives a penalty. There are also exceptions for each case, which are described in the following graph.

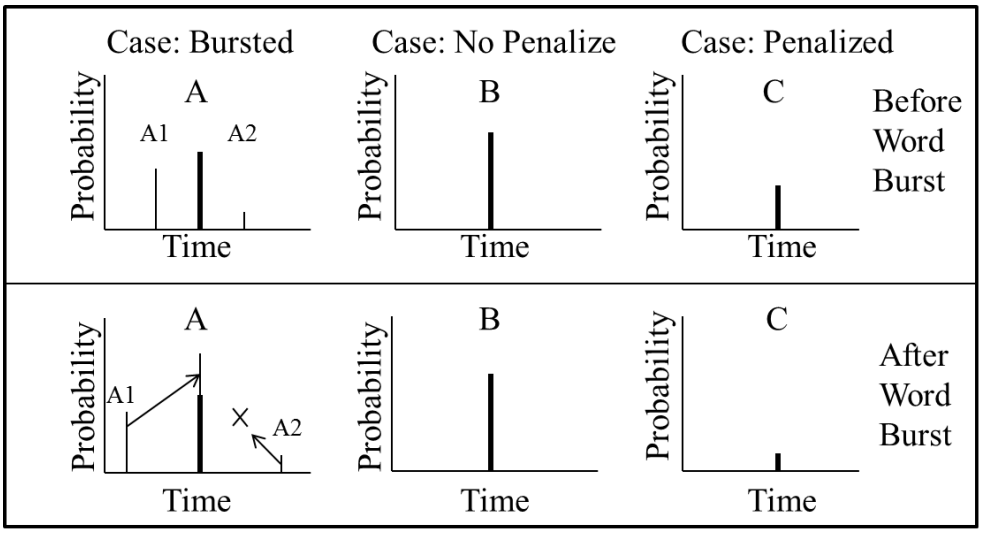

Figure 4.3: Concept of Word Burst Rescoring

Figure 4.3 illustrates Word Burst rescoring. The y-axis in this figure shows the probability of each hypothesis. There are three cases A, B, and C to which we apply our rescoring algorithm. There are A1 and A2 that are the same word as A. Both of them occur near A in time. Both B and C have no nearby instance of the same word.

The transition in the figure shows how the rescoring algorithm works. There are three possible cases that are represented by hypotheses A, B, and C for Word Burst Rescoring. For hypothesis A, the probability of A is boosted in proportion of A1's 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 hypotheses 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 uses probabilities 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 the decoding result. We assume that very low p means an error.
- Some words do occur alone. These words can be assigned very high probability during decoding. Penalizing all isolated word harms performance, so we do not do it.

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:

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

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

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

where $w(x_i, x_j)$ is the weight for Word Burst between x_i and x_j, and S is the sum of all weights for different 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, $penalty(L)$, and $windowsize$ are computed from the development set.

Performing extra iterations of Word Burst rescoring does not achieve further improvement; as a single iteration will significantly affect the probability distribution of each word in the hypotheses, the rescored recognition hypothesis already has an extreme confidence for each hypothesis, and the false alarm that would get penalized is already been penalized significantly. We tried on our rescored result and none of them can benefit again from iterative application of the same rescoring algorithm.

Target Extension for Word Burst

Word Burst uses the recurrence of a word hypothesis for rescoring but relies on the recurrence of the same word, thus is a problem in agglutinative languages where the hypothesis may reoccur, but as a morphological variant. This phenomenon limits the detection of Word Bursts. Section 4.6.1 presents describes experiments on extension.

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 theme of the thesis. We present two different approaches to address this problem.

The first approach is 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.

The second approach is 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.

4.3.2 Unique Penalization Rescoring

Unique Penalization rescoring focuses on the second assumption we presented at the beginning of the chapter; a word that appears alone without other identical instances in close proximity tends to be recognition error. In this set of experiments, we wish to identify: What is the appropriate "close approximate"? The rescoring algorithm here is really simple: If a recognition hypothesis only appears in the designed size of context once (unique), we consider it as a recognition error, and hence reduce its recognition confidence score. In this section we examine the proper context size that can support our assumption. The experiments are reported at section 4.6.2.

Three levels of Context

Since we are working with a recognition hypothesis from an ASR system, only the hypothesis that has a high probability from the ASR system can be considered as a word that is really present in the corpus. The corpus we use here will be described in section 4.5.1, which has 10 hours of speech in six different languages. Unique Penalization 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 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 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.

Since the data we have consist of conversational telephone speech, it is limited to two speakers at most. The conversation context can be considered as twice the size of speaker context. The corpus context represents the ASR system's tendency to some extent, as it relies on whether some of the words have ever been decoded confidently by the ASR system. If a word does not have any instance of hypothesis that is recognized confidently, it is possible that the word really did not exist in the corpus, or the ASR system does not tend to recognize that word.

Algorithm Development Process

After realizing the most of the gain from Word Burst rescoring coming from the false alarm reduction (more detail will be discussed in section 4.7.1) in penalization step, we decide to further investigate how to perform the penalization step in a better way. The problem with the predefined window size is that it might require tuning to get the appropriate context size, we wish to use natural boundaries to investigate the penalization part of the Word Burst Assumption. Since we are working on conversational telephone speech, each phone call (conversation) is a natural boundary, and if we consider each speaker may have his or her own set of word they will use, separating different speakers becomes another context level (Speaker). If the data we worked on has topical label of the conversation, it might be a good context size for multiple conversations in the same topic. However, since that information is not available, we decided to use the entire corpus as the largest context level.

The parameter tuning process is similar to how we did in Word Burst, yet since this algorithm only has the penalization to tune, the process is simpler. We tune the penalization parameter that has the best performance on the conversation level and applied that to all three different levels. The reason why we pick all three levels using the parameter from conversation level is because conversation level already shown the best performance on development data, and we expect the testing data might have similar trend.

Rescoring Algorithm

Unique Penalization uses the context we described in the previous section for hypothesis rescoring. According to our Unique Penalization assumption, we can partition all decoder output 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.

Unique Penalization relies on "high confidence words" to prevent the word from being classified as a recognition error. The "high confidence word" is decided by a threshold. The threshold is the average of the top 50% highest posterior probabilities for a hypothesis per utterance.

Hence, the threshold changes according to different utterances. This dynamic threshold can preserve a "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 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 Unique Penalization Rescoring. $p(w)$ is the original confidence of score before 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 Unique Penalization Rescoring.

4.3.3 Difference between our approaches and previous work

Both Word Burst and Unique Penalization capture the relationship between the repeated words during conversation. Word Burst focuses more on manipulating the recognition hypothesis in a smaller temporal window, while Unique Penalization focuses on varying the size of context at three different levels to determine which is the most ideal setup for conversation speech. Both approaches follow the similar intuition with the previous work on exploiting context for language processing, yet there are a few major differences. First, most of the work done before (Kuhn and De Mori, 1990; Kupiec, 1989) relies on a model for the content words, while our approach only requires a limited amount of text for deciding a stop word list. Our approach does not require knowledge/data on the content word (Jelinek et al., 1991) to leverage it; this makes it easy to adapt to new languages. Second, the sources of context are different between our approaches and the traditional context works. The earlier work obtains context from a sizable number of text documents or well-structured spoken documents (Church and Gale, 1995), where the source of context is of good quality and quantity. Our approach employ the context from the noisy decoder output, and only the words within few seconds of each utterance. The quality and quantity of the available context is not as good as in previous work, yet it can be leveraged with far fewer requirements. In addition, Word Burst utilizes the context of temporary topics during conversations, which is dependent on the time distance instead of the number of recognized word tokens (Rosenfeld and Huang, 1992). This captures the degradation of specific topics regarding times, since the silence during conversation leads to the end of a topic. While the traditional work focuses on the token distance or whether it is in the same document, it does not put any focus on time and silence. A 20-second silence is ignored in the previous works, yet it can mean the end of current topic in the Word Burst setup. (Young et al., 1989) only considers the temporal distance between different utterances, while our work also addresses the temporal distance within each.

The difference on strategy of leveraging Word Burst in our current work and previous work also demonstrate the difference of Word Burst on text document (such as article or web page) and conversational speech. First, the text version's (Church and Gale, 1995) burst range is the entire

document, while the conversational version has a more limited window for about 20 seconds. This differences indicates the fact that it is more likely to have topic change in conversational speech than in text documents, so the coverage of bursts in the conversational setup is more limited. Second, the conversational version requires removed of stop words before applying Word Burst. There are lots of meaningless word could be spoken in conversation, while the text document usually are more clean. Third, the way we leverage Word Burst on conversational speech includes both a bonus and the more likely content word and a penalty for the potential errors, while in the text setup is focusing on the bonus only. Since the earlier Word Burst was introduced in the Information Retrieval community, it focused mostly on text documents instead of ASR results. The reduction of false alarm, which is the main contribution for our work, is not the focus of the earlier research on textual Word Burst, as it does not expect errors to show up in the textual document. We believe the Word Burst exists in these different form of data, yet the details are different, so we need to leverage it with different ways.

4.4 Word Burst for Identifying Recognition Errors in Conversational Speech

A potential drawback of Word Burst rescoring is that it requires intensive tuning to make it effective. Since Word Burst rescoring benefits STD by reducing false alarms, we assume the Word Burst should be a good feature for identifying recognition errors. In this section, we describe an approach based on Conditional Random Field (CRF) (Lafferty et al., 2001) modeling to leverage the Word Burst phenomenon as a feature for identifying potential recognition errors. We demonstrate that, even without intensive parameter tuning on development data, Word Burst can be used to improve the identification of potential recognition errors. We include this set of experiments to demonstrate the generality of the Word Burst as a conversational feature, we moreover show it can be leveraged without tuning on development data. The experiments are reported at section 4.6.3.

CRF based sequential labeling for recognition errors

Lafferty et al. (Lafferty et al., 2001) introduced CRF, a framework for building probabilistic models to segment and label sequence data. It proved to be a successful approach to many Natural Language oriented sequential labeling tasks (Finkel et al., 2005; Jiang, 2005; Sha and Pereira, 2003). We describe our strategy of using CRFs for labeling recognition errors.

Given the recognition hypothesis from a specific ASR system (and the reference, for identifying the different types of recognition errors), we first compute its WER, and identify each type of recognition error. From the recognition hypothesis, we can assign each word a label, depending on whether it is an insertion error or a substitution error. Note that, since we are only labeling the word that shows up, our approach does address deletion errors. Once words are given a label of either recognized correctly or not, we can train a CRF model. For subsequent decoding by the same ASR system, we perform sequential labeling from the CRF model we just trained. To leverage our Word Burst as a feature for labeling recognition errors, we give each word an extra dimension of feature which contains the Word Burst flag that could be True, False or Stop. This

training process does not require grid search for different parameter for the model, which we believe is less ad-hoc.

Word Burst as a language independent feature

Multiple features have been proposed to estimate ASR confidence (Benıtez et al., 2000; San-Segundo et al., 2001), these features include different forms of acoustic scores from the ASR system, the n-best list of recognition hypothesis, durations and language modeling scores. Most of these works were focusing on the information from the ASR system, while the environment and setting when the speech occurs is not considered. For example, there are no distinction between processing a scripted recording or conversational telephone speech or broadcast news.

We leverage knowledge about human conversational speech to support our task. Word Burst, a phenomenon we observed in conversation is an ideal candidate. Word Burst is only useful for content words; common words recur for various reasons. Since we are not planning to introduce any language specific information, we assume that the most frequent 1% words in our data are stop words for that language. As a result, for each word, there could be three options for Word Burst based features: True, False or Stop. True or false indicate whether the Word Burst occurs for this word in the same utterance and the Stop indicate the word belongs to stop word list. We also consider the Word Burst feature for the word before and after the current labeling word, in order to provide richer contextual information.

This feature has some benefits. First, it can be applied to any language, as it only needs the occurrence pattern in a conversation, it does not requires any language specific knowledge. Also, since it does not rely on ASR system specific information (just the output), it can be used in different systems. As well, reliance on the properties of conversation, means it can be applied to recognition in any language.

Since the purpose of this work is to validate the Word Burst feature, there is no algorithm development or parameter tuning process for this work. We simply use whether there are multiple instance of the same word in a single sentence to represent word burst as a feature in standard CRF sequential labeling, no parameter/algorithm change here, only one additional feature.

4.4.1 Difference between our approaches and previous work

Our focus on Word Burst, or the conversational context, is the main difference between our work and the traditional CM works. While identifying new features, such as using phonetic recognition result comparing with word recognition result, is already an approach that had been reported in several papers (Benıtez et al., 2000; San-Segundo et al., 2001), these features only focus on the ASR system side, and consider nothing about the context of the conversation. Our approach is based on we knowing the recording is a conversational speech, hence the characteristic of conversational speech can be leveraged within this task. We use recurrence of the identical recognition hypothesis within close temporal distance as feature, which is also new compare to the more common n-gram based approach. It's about introducing a new feature that's dedicated to a specific type of recording.

4.5 Speech corpora and experimental design

In this section we will introduce the detail for our experiments, including the dataset we use in both experiments, the experimental setup, the evaluation metrics and the tool we used. For the STD under Limited Resource Condition, we also describe the distinction between two different query sets, since these sets have different characteristics.

4.5.1 Dataset

For the rescoring experiment, we use six different conversational telephone speech recording dataset from six different languages: Cantonese, Pashto, Tagalog, Turkish, Vietnamese and Zulu, as provided by the IARPA BABEL program (Karakos et al., 2013; Mamou et al., 2013). For the Word Burst target extension experiment, we focus on Zulu and Turkish, since we wish to confirm the effect on multiple agglutinative languages. 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. The tuning is done using grid search over potential values of each parameters. For the final result, we used the parameter that achieved the best performance on the development data.

For the identifying recognition error experiments, we run our experiments on five different datasets, including two English datasets with different WERs, two Tagalog datasets with close WER but different lexicon size, and a Zulu dataset. The reasons for selecting these languages are: English is the language has the best overall ASR performance among these languages. However, even on English, it can still lead to relative high WER, hence our English experiments can describe the limits of our approach on such data. The Tagalog experiments show the limitation on a low resource language. When the WER increases due to the limited amount of training data available, there is also another limitation on the vocabulary size. Both system use the same acoustic model for decoding, yet using different language model with differernt vocabulary size. Zulu is in the same space as Tagalog, yet being an agglutinative language also makes Word Burst detection difficult. For the English dataset, we use YouTube "How To" videos (Yu et al., 2014) which has a correct transcript but which was also artificially degraded to 20% WER and 40% WER. This was also used in chapter 6.

4.5.2 Experiments setup

STD system description

Our STD system uses an ASR-search two-stage pipeline. The decoded hypotheses are represented as confusion networks, which is the default setup of our recognizer. Confusion networks are generated from the combination of three different decoding systems (Finke et al., 1997). Both Word Burst and Unique Penalization Rescoring are applied to every hypothesis in the confusion network. Our search component outputs the location of queries in the confusion network, but it skips OOV queries and only retrieves results for IV queries.

Recognition error classification system description

We conducted our experiments using five different setups and in three different languages, using several open-source toolkits. We use the CRFSuite toolkit [1] with its Python wrapper python-CRFSuite [2]. For the L1 and L2 coefficient penalty, we used 1.0 for L1 and 1e-3 for L2, recommended for building a NER system on CoNLL 2002 data (Tjong Kim Sang and De Meulder, 2003). For creating the training data for the CRF modeling (which includes labeling which word is incorrect in the decoded result), we use the Python toolkit asr-evaluation [3] which labels substitution and insertion errors for the ASR result. This serve our purpose better as it provides more detailed information about errors in the decoder output. For the evaluation metric, we report F1 score, along with precision and recall.

Evaluation Metrics

The evaluation metrics for STD is introduced in section 1.3.1. In this chapter, we focus on the ATWV metric, which is computed from the mean of every query's TWV score with a fixed threshold for each query. This is the main evaluation metrics within the IARPA BABEL program. One issue of the ATWV score is that it is really hard to interpret the result. Hence, we will also provide IR based metric including precision, recall and F1 score. Note that one of the main difference between IR metric and TWV based metric is that, for the IR metric, there is no special weight on correct or error detection, yet within the TWV based setup, the correct detection and false alarm has different weights.

For identifying recognition errors, we report standard classification result including F1 score, precision and recall.

Code

We had published the code that had been used to completed the experiments on github. For the experiments we described in this section, the Word Burst repository [4] contains the code for both rescoring experiments, and the other repository [5] contains the code for identifying recognition errors with Word Burst feature.

Original query set and Non-singleton query set for STD

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. Both of our approaches 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

[1] https://github.com/chokkan/crfsuite
[2] https://python-crfsuite.readthedocs.org/en/latest/
[3] https://github.com/belambert/asr_evaluation
[4] https://github.com/jltchiu/Word-Burst
[5] https://github.com/jltchiu/Recogntion-Error-Classification

Table 4.2: Singleton Query Distribution in different test languages

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

Table 4.3: ATWV for Word Burst Rescoring with original query set

Language	Baseline	Word Burst	$\triangle$ 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
Zulu	0.115	0.122	+6
Mean	0.127	0.142	+12

listed in Table 4.2. A 45% singleton query rate means that 45% of the queries occur only once in the testing data. Accordingly, we also remove all singleton queries and present non-singleton query set results separately.

4.6 Experimental results

4.6.1 Result: Word Burst Rescoring

Our experiments are conducted on six languages. 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 the effect of Word Burst rescoring, since the differences between query sets are eliminated. The query with one one occurrence in the entire corpus does not have "other instance of the same word" to trigger our rescroing. The IR metric we reported also make our results more interpretable. The approach is introduced at section 4.3.1.

From the result shown in Table 4.4 and 4.6, we can see our improvement is mostly on better precision for the detection. We will discuss more in the Analysis session. Comparing Table 4.3 with Table 4.5, we can observe that the improvements in non-singleton queries are larger and more stable. This indicates that Word Burst is more useful in the condition without singleton queries, because the word that shows up multiple times in conversation tend to be the one that has topical meaning and can be observed with Word Burst phenomenon. The Vietnamese has the largest ATWV difference between the original query set and the non-singleton query set, since it has the highest singleton queries percentage. The Turkish performance is still limited because it

Table 4.4: IR Metrics for Word Burst Rescoring with original query set

Language	**Baseline**			**Word Burst**		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Cantonese	0.47	0.20	0.28	0.53	0.18	0.27
Pashto	0.48	0.26	0.34	0.57	0.25	0.35
Tagalog	0.52	0.33	0.40	0.62	0.33	0.43
Turkish	0.65	0.30	0.41	0.64	0.30	0.41
Vietnamese	0.51	0.13	0.21	0.54	0.14	0.23
Zulu	0.44	0.20	0.28	0.49	0.21	0.29
Mean	0.51	0.24	0.32	0.57	0.24	0.33

Table 4.5: 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
Zulu	0.138	0.153	+11
Mean	0.125	0.144	+15

Table 4.6: IR Metrics for Word Burst Rescoring with non-singleton query set

Language	**Baseline**			**Word Burst**		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Cantonese	0.54	0.21	0.30	0.60	0.19	0.29
Pashto	0.54	0.27	0.36	0.62	0.26	0.37
Tagalog	0.57	0.33	0.42	0.66	0.34	0.45
Turkish	0.70	0.30	0.42	0.69	0.30	0.42
Vietnamese	0.58	0.14	0.22	0.61	0.15	0.26
Zulu	0.53	0.18	0.27	0.57	0.19	0.28
Mean	0.58	0.24	0.33	0.63	0.24	0.35

Table 4.7: Paired t-test result for Word Burst Resorcing

Language	Mean	t	p	n
Cantonese	+0.05	17.4	<0.001	3018
Pashto	+0.09	33.6	<0.001	6830
Tagalog	+0.04	11.3	<0.001	3188
Turkish	-0.03	-22.3	<0.001	899
Vietnamese	+0.03	18.7	<0.001	5260

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

is an agglutinative language, the word in Turkish could reoccur in conversation as morphological variant, we conduct target extension experiments to deal address this issue.

We performed a paired t-test to confirm the improvement we obtained from our approach is statistically significant except for Turkish, which does not have an improvement on the performance but shows a significant decrease in the mean. The result is presented at Table 4.7. Also, the Mean of Turkish changed in the different direction from other languages. It is because Turkish is an agglutinative language, the word is less likely to occur in the exact same form. As a result, the penalization threshold is very high, otherwise it will over penalize the result and hurt ATWV. For other languages, the major change and gain on performance is on the penalization part, where in Turkish, the main change is on the burst part and less effective compare to the penalization. This result in our approach does not have significant improvement for Turkish.

Result: Word Burst Target Extension

We examined 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 focus this on the original query set since we expect that the target extension can overcome singleton queries by extension from the non-singleton queries. The purpose of target extension is to make the approach useful even with singleton queries. The approach is introduced at section 4.3.1.

Table 4.9: IR metrics comparison between target extension approaches

Language	Baseline			SubString			Morphology		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Turkish	0.65	0.30	0.41	0.67	0.30	0.42	0.66	0.30	0.41
Zulu	0.44	0.20	0.28	0.49	0.21	0.29	0.48	0.20	0.29
Mean	0.55	0.25	0.35	0.58	0.26	0.36	0.57	0.25	0.35

Table 4.10: ATWV for different levels of 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 4.11: F-score for different levels of context with original query set

Language	Baseline	Speaker	Conversation	Corpus
Cantonese	0.28	0.27	0.27	0.28
Pashto	0.34	0.33	0.34	0.34
Tagalog	0.40	0.39	0.39	0.40
Turkish	0.41	0.41	0.41	0.41
Vietnamese	0.21	0.21	0.21	0.21
Mean	0.33	0.32	0.32	0.33

Tables 4.8 and 4.9 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. And the gain is mostly obtained by achieving better precision. Interestingly, a simple procedure that uses an orthographic substring match works better than the morphological decompositions we used. There is more analysis for target extension in the Analysis section.

4.6.2 Result: Unique Penalization Rescoring

The Unique Penalization Rescoring experiments are conducted on the same five languages as Word Burst Rescoring. For each language, we perform the experiment using all three levels of 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 Unique Penalization, since the differences in query sets are eliminated. The approach is introduced at section 4.3.2.

Tables 4.10 and 4.12 compare three different levels of context in five different languages, with two different query sets. Unique Penalization Rescoring produces limited improvement on Turkish, because it is an agglutinative language. It is harder for the same word to show up multiple times in an agglutinative language, so the penalty will be applied too often if the threshold is not high. However, if the threshold is high, then the penalty will be hard to trigger, hence lead to limited improvement. The rescoring algorithm for Unique Penalization does not affect morphological variants of words. The Corpus context shows the least improvement. We discuss the reasons for this in the Analysis section. The Speaker and Conversation are more appropriate levels of context. These levels of context provide similar improvement in ATWV, although Con-

Table 4.12: ATWV for different levels of 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

Table 4.13: F-score for different levels of context with non-singleton query set

Language	Baseline	Speaker	Conversation	Corpus
Cantonese	0.30	0.28	0.29	0.30
Pashto	0.36	0.35	0.35	0.36
Tagalog	0.42	0.41	0.41	0.42
Turkish	0.42	0.42	0.42	0.42
Vietnamese	0.22	0.22	0.22	0.22
Mean	0.34	0.34	0.34	0.34

versation provides more consistent improvements. The improvement we achieved on Speaker and Conversation setups are statistically significant in pair-wise t-test ($p < 0.01$) between the baseline and Speaker, and the baseline with Conversation. The t value and mean/standard deviation difference here follows the same trend as the t-test reported in the Word Burst experiments (section 4.6.1).

Tables 4.11 and 4.13 shows the F-score of the results. We observed that, despite having improvements on ATWV with Unique Penalization, the F-score does not increase as was observed in Word Burst Rescoring. We believe the reason is because F-score does not apply different weights on correct detection and errors, while ATWV did. Our modification has very limited effect when the weight of every instance are equal, but still can have an impact in the ATWV-based setup.

In Table 4.14, we show the relative improvement in ATWV on the Conversation level using different query sets. The improvement on the non-singleton query set is more consistent and

Table 4.14: 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

Table 4.15: Tagalog High Vocabulary Size (WER81)

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.59	0.35	0.44	7552
Error	0.59	0.79	0.68	8915
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.61	0.38	0.47	7552
Error	0.60	0.79	0.68	8915

Table 4.16: Tagalog Low Vocabulary Size (WER84)

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.58	0.19	0.28	5557
Error	0.64	0.91	0.75	8790
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.57	0.19	0.28	5557
Error	0.64	0.91	0.75	8790

higher. Cantonese and Vietnamese have the most distinctive difference between the original query set and the non-singleton query set. This difference is due to the high singleton query percentage in the original query set. By eliminating the difference in the query set, we can observe unbiased performance improvement on ATWV with Unique Penalization Rescoring.

4.6.3 Result: Identifying Recognition Errors

Tables 4.15 to 4.19 show the recognition error classification result on five different setups, the WER of dataset is on the title of each table. By including Word Burst as a feature for our classification task, we achieved statistically significant (with pair-wise t-test and $p < 0.05$) improvement on either labeling correct or error in 2 out of 5 datasets, which are the Tagalog High Vocabulary and YouTube WER40 dataset. There are three major conditions that will limit the effectiveness of our features: WER of the decoder output, limited vocabulary size and agglutinative languages, the detail of these limitation will be discussed in the Analysis section. The approach in is introduced at section 4.4.

Table 4.17: Youtube (WER40)

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.73	0.94	0.83	72566
Error	0.64	0.23	0.34	32175
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.74	0.92	0.82	72566
Error	0.62	0.29	0.39	32175

Table 4.18: Youtube (WER20)

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.85	1	0.92	92686
Error	0.59	0.02	0.03	16837
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.85	1	0.92	92686
Error	0.57	0.02	0.03	16837

Table 4.19: Zulu (WER81)

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.58	0.25	0.35	3292
Error	0.76	0.93	0.84	8372
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.57	0.22	0.32	3292
Error	0.75	0.93	0.83	8372

4.7 Analysis

4.7.1 Tradeoffs between Correct Detections and False Alarms with Word Burst rescoring

Table 4.20: Tradeoffs between Correct Detections (CD) and False Alarms (FA) in Unique Penalization Rescoring and Word Burst Rescoring (Change in %)

Language	**Unique Penalization**		**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

Table 4.20 shows that both Unique Penalization Rescoring and Word Burst Rescoring contribute to more than 25% of false alarm reduction in three languages. The major difference is that Unique Penalization Rescoring tends to "overkill" correct detections while Word Burst rescoring has a better mechanism to preserve the correct detection. The threshold for deciding whether to apply Word Burst rescoring avoids removing a correct detection, yet sacrifices some of the false alarm reduction power. The value of the threshold was decided by tuning on the development data with various parameter choices. However, in our evaluation setup, this value favors correct detection much more than reducing false alarms. This reflects on the observed ATWV difference, in that Word Burst rescoring generally has better ATWV improvement compared with Unique

Penalization Rescoring. Although Word Burst rescoring resulted in fewer false alarms, correct detection contributes to a higher ATWV score. The source of improvement for both approaches are similar: reduction in false alarms.

The only exception in Table 4.20 is Turkish, which has lesser improvement compare to every other language. Turkish is an agglutinative language. Hence, the word usually reoccur in in morphological variants, making identical words showing up less frequent. The target extension is our effort to let the morphological variants can still be captured. The correct detection / false alarm trade-off for target extension is shown in the Table 4.21, where we compare Turkish to Zulu, also an agglutative language.

Table 4.21: Correct Detection/ False Alarm tradeoff for two target extension approaches with Word Burst Rescoring (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 4.21. This indicates that target extension can restore 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 languages.

4.7.2 Applying our approaches on better-quality ASR results

We also tried our approach on data with lower WER. For the four languages on which we conducted our experiments, we also have another ASR system that is trained on 80 hours of data (compared with 10 hours of training data we presented in our Experiments section). This setup is closer to STD under the Rich Resources Condition compared with our Limited Resources Condition. The WER on 80 hours of training data ranges from 50% to 55%, while the WER for 10 hours of training data ranges from 60% to 70% for all of these four languages. Our approach does not provide as much improvement as the result we presented in Table 4.22, even with the newly tuned parameter. The ideal $penalty(L)$ we obtained from development data is very close to 1, which means that it is best not to perform any penalization when it does not match our assumption. Since our main improvement on the Limited Resources Condition comes from the reduction of false alarms, having the $penalty(L)$ close to 1 indicates that we are not going to achieve as much false alarm reduction.

This indicates the characteristic of our approach. If the ASR performance is already of good quality, applying the assumption from Word Burst cannot always achieve better performance. The reason behind it is cause we are still applying an assumption to data, and the assumption will always have exceptions. When the exceptions happens, the rescoring algorithm will damage the performance. Assuming that there is a perfect ASR recognition hypothesis available, any

Table 4.22: Word Burst on 10 hours (High WER) and 80 hours (Low WER) of training data

Language	10hr Base	Word Burst	△ ATWV (%)	80hr Base	Word Burst	△ ATWV (%)
Cantonese	0.114	0.122	+7	0.322	0.324	+1
Pashto	0.073	0.103	+41	0.214	0.221	+0
Tagalog	0.136	0.173	+27	0.358	0.359	+0
Turkish	0.241	0.245	+2	0.385	0.383	-1

rescoring will potentially damage the quality of the recognition hypothesis unless it is not doing anything. After all, all assumptions has exceptions, and the way we apply Word Burst is to assume everything follows a specific structure. It can clean up parts of the data when they are noisy, yet it will introduce extra errors when high-quality data are already available. This result emphasizes that maximum benefit of these techniques will be found under conditions of limited resource or poor recognition performance, emphasizing that structural properties can compensate for poor baseline performance.

4.7.3 Words classified as errors in Unique Penalization Rescoring

Table 4.23: Percentage of words being classified as errors at different levels of 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 4.23 shows the percentage of words being classified as recognition errors in different setups. These words are the main focus for Unique Penalization Rescoring. Except Turkish, all languages exhibit similar trends. Turkish is the exception to this pattern due to its morphological variation. The variation reduces the performance for classification of recognition error, 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 important observation: Defining a large context that includes too many confidently recognized words weakens Unique Penalization Rescoring. The Corpus level of context classifies most of the words in the corpus as being recognized correctly. Consequently, Unique Penalization Rescoring only works on a small portion of data, and does not sufficiently impact ATWV. We recommend that most of the time Word Burst rescoring is going to provide better and more consistent improvement over Unique Penalization.

4.7.4 Unsuccessful Word Burst target extension

In addition to the work we presented in Section 4.3.1, we also investigated other possible approaches to target extension. Since Word Burst relies on context, we examined other sources of context-based 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 incorporate this MI information in the rescoring process. The words that were observed to co-occur in the training data receive a bonus, while the words that never co-occur in the training data are penalized. This did not work, as the MI we computed is from a limited training corpus (10 hours). This approach harms correct detections, since the co-occurrence distribution is different from that of the training data. In order to make it works better, we expect a bigger corpus for computing MI is necessary. The MI computed from more data can better represent the co-occurrence of different words, and that can be a better trigger for Word Burst rescoring.

We also examined Brown clustering (Brown et al., 1992) on the training corpus. Brown clustering places all words in the training corpus into several clusters. The algorithm groups items into classes, using a binary merging criterion based on the log-probability of a text under a class-based language model. As a result, the output can be thought of not only as a binary tree but perhaps more helpfully as a sequence of merges, terminating with one big class of all words. We extend the Word Burst target to other words in the same cluster. This does not improve ATWV, and is likely due to the fact that limited training data cannot create high-quality clusters.

We investigated LDA-based topic modeling (Blei et al., 2003) for target extension. The assumption is that since we claim that the recurrence of the word is caused by the ongoing discussion topics, if we can perform Word Burst based on the word that belongs to the same topic, it should have positive results. Our topic model is trained on 80 hours of transcribed speech in Tagalog, since a good topic model requires a reasonable volume of data to train, and 10 hours of transcribed speech is very limited. We trained with two different topic counts, 10 and 30. We then computed topic distributions and selected the top X words from each topic. Among these selected words, each word has target extension to other selected words in the same topic. Since this is an experimental setup, we focus on Tagalog, as it has a reasonably good improvement for the standard Word Burst.

Table 4.24 shows the result for Word Burst target extension using word clustered from LDA topic modeling. None of the target extensions with topic modeling outperform standard Word Burst. When the numbers of the top X words we selected from each topic is low, it achieved the same performance as standard Word Burst. After examination, this is because the target extension is never triggered, and it just performs identically with standard Word Burst. When the top X words is set to a higher value, the target extension starts to happen and the performance starts to drop. This indicates that it is very difficult to provide gains outside of identical words using target extension. (Rosenfeld and Huang, 1992) described a similar finding in their work with "self triggering".

Last but not least, since word embedding achieved good improvement on our work in Chapter 6, we also attempted to use word embedding as a way to expand our Word Burst target, and the result is reported in Table 4.25. We created word embedding models in Tagalog according to the

Table 4.24: Extending Word Burst with Topic Models

Number of Topics	Top X Words	ATWV
Baseline		0.136
Normal Word Burst		0.173
10	10	0.173
10	20	0.173
10	100	0.170
10	200	0.166
30	10	0.173
30	20	0.170
30	100	0.169
30	200	0.165

80 hours data we have in FullLP set, and aid clustering based on the location of the word in the word embedding spaces. We expand the Word Burst target for every word form other instance of itself to every other word belongs to the same cluster. The result is similar to the one obtained using our LDA target expansion. The more targets each word can trigger Word Burst, the worse the STD result will be. As a result, it is still difficult to use related word to perform Word Burst. If we have an way of modeling semantics for every words, Word Burst with identical semantics might be the first step go beyond identical words. Since that will be a closer approximation of using on identical words, compare to related words.

Table 4.25: Extending Word Burst with Word Embedding

Number of clusters	ATWV
Baseline	0.136
Normal Word Burst	0.173
10	0.073
30	0.118
50	0.135
100	0.145

4.7.5 Substring-based Target Extension on Other Languages

We also conducted experiments for substring-based target extension for Word Burst on non-agglutinative languages. For a language like English, if two different words has matched substring, then we consider Word Burst happened. For example. the word “interest” and “internet” will be considered as Word Burst since they share a substring of “inter”. The experiments show that target extension on non-agglutinative languages does not outperform regular Word Burst rescoring. If we set the length of a substring too short, target extension triggers too many words, leading to false alarms. On the other hand, if we set the length of a 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. Table 4.26 shows the result on Tagalog.

Table 4.26: Substring-based Target Extension on Tagalog

Length of substring	**ATWV**
Baseline	0.136
Normal Word Burst	0.173
3	0.081
4	0.082
5	0.095
6	0.122
7	0.123

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.

4.7.6 Analysis on Identification for recognition errors with Word Burst

WER related analysis

The two experiments on the YouTube data demonstrate how WER affect the effectiveness of the feature. When the WER is already low, training the CRF with a low WER data can not learn the characteristics of the recognition error well, as the number of instance for correct and error labels are also very different. (92686 correct vs 16837 error). However, when the training data WER increased, the feature starts being effective, because there are more instances to train, and also the distribution of labels is more balanced. With this observation, we also attempt to use the CRF model trained on WER20 YouTube data to test on the WER40 YouTube data.

Table 4.27 shows the result of training the CRF model on the WER 20 data and then test it on the WER 40 data. There is no significant difference on the performance of the classification result. This indicates the model trained on WER 20 data is not robust enough even if applied to WER 40 data. However, when the model is trained on the training data that has more recognition errors, our feature can start being effective. Because the model trained with WER 40 data has balanced training example compare to WER 20 data, which has less error examples.

Vocabulary size related analysis

The two experiments on the Tagalog data demonstrate another important factor for the feature to work: it requires reasonable vocabulary size for the ASR system. Both setup has similar high WER (81 vs 84, shown in Table 4.15 and 4.16), yet the difference on vocabulary size for the language model is significant (21098 vs 5565). Although having a bigger vocabulary size does not guarantee a better WER, it brings more variety on the training data for the CRF model. The

Table 4.27: YouTube WER20 model on WER40 data

Baseline	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.70	1	0.82	72566
Error	0.71	0.01	0.03	32175
Word Burst	**Precision**	**Recall**	**F1**	**Instance**
Correct	0.70	1	0.82	72566
Error	0.65	0.02	0.03	32175

distribution of the recognition errors on both training and testing data for the CRF model also shows this phenomenon.

Table 4.28: Error Distribution on Tagalog setups for Identification of recognition errors

	Insertion	**Deletion**	**Substitution**
Low Vocabulary Size-Test	1609	12575	7181
High Vocabulary Size-Test	2647	11493	6268
Low Vocabulary Size-Train	4858	35255	2932
High Vocabulary Size-Train	7693	32059	2189

Table 4.28 shows the difference on the distribution of recognition result on two different datasets. The Low Vocabulary Size setup has a smaller vocabulary size. Hence, their decoding result contains more deletion errors; since there is less potential approximate for some of the words in the speech data. Note that for training the classifier for recognition errors, we can only identify insertion errors and substitution errors. Having a bigger vocabulary size can increase the amount of insertion and substitution errors, which makes more errors being able to be detected by our approach.

Agglutinative Language related analysis

The experiment on Zulu setting is proposed as a comparison with the experiments with Tagalog High Vocabulary Size system. Although the ASR result on both languages has 81% WER, the effectiveness of the Word Burst is very different. Earlier work (Chiu and Rudnicky, 2013) reported that Word Burst will have limited effectiveness on agglutinative language, since words may to recur in other morphological form in those languages. Our experiments here suggested similar results. We believe the key to make Word Burst useful for agglutinative languages is to find a way to group every word's morphological variants in the same group. The ultimate version of this grouping will involve putting semantically similar words into the same group. However, this is still an open problem even in English, because there is not a fixed set of semantic labels that can be used in every situation. It is even more challenging if we are trying to address this problem with a language independent approach. We believe the word that is spoken in the conversation is only a token to express the semantics, and if we are able to capture the semantics, the different word tokens that are used to provide same semantics should be treated equally.

Translating Identification result into WER

One common question that follows from our better classification result is that: "Even if we are able to identify the recognition better, can we convert this result into better WER"? We tried to remove all of the word that we classified as recognition errors to see whether that will improve the WER. The assumption is that, if we can remove enough insertion errors, then the WER can be improved in this way. However, insertion errors are usually not the primary source of recognition errors, and removing all words that's been labeled as recognition errors will also remove some correct words, it does not provide any improvement. As a result, we think the contribution of our work is more on identifying what are the potential recognition errors, and has less to do with fixing/recovering it.

4.8 Discussion

4.8.1 Contribution from this Chapter

The general contributions we made in this chapter are:

- We identify a Word Burst phenomenon that occurs in conversational speech. Word Burst describes the phenomenon in which a word that has been spoken recently in conversation is more likely to recur in close proximity. (4.3.1)
- We verify that our assumptions can be applied to multiple languages, which indicates that Word Burst is not a language-dependent phenomenon. (4.6.1)
- We demonstrated that leveraging Word Burst can efficiently reduce or classify the noise in the communicated information when the delivered information is noisy. (4.6.1)
- We also demonstrated that, in order to leverage Word Burst, it does not require a large amount of data to train a model. Good improvement in performance can be achieved by simply applying our knowledge to processing data. (4.6.1)

The task-specific contributions we described in this chapter are:

- We designed two different rescoring algorithms, Word Burst rescoring and Unique Penalization rescoring, that improve STD under Limited Resources Conditions. We demonstrate that these algorithms work for multiple languages. (4.3.1, 4.3.2)
- The rescoring algorithms we presented do not require large amounts of data to train the model, which makes them easy to deploy to different languages. (4.6.1, 4.6.2)
- The effects of the presented algorithms are mostly related to false alarm reduction, these rescoring algorithms can thus provide a simple way of cleaning up recognition errors. (4.7.1)
- We described a target expansion technique for our rescoring algorithm to allow the effect to extend to agglutinative languages, such as Turkish and Zulu. (4.3.1)
- When selecting the size of context for context-based processing, conversation is the best sized unit of context to leverage, as opposed to using the entire corpus which combines many different contexts. (4.3.2)

- We determined that our proposed approach has limited effect on cleaner data. (4.7.2)
- We also identify how the incidence of the query term affects the effectiveness of the rescoring algorithm in STD. If the query only occurs once in the testing data, then the rescoring algorithm relying on the same word will not produce the desired effects. (4.6.1)
- We demonstrated that, Word Burst can also contribute to other tasks such as identifying recognition errors by using it as a feature. (4.4)

4.8.2 Unresolved Issues

In addition to what we already presented in this chapter, we also identify several potential issues that could be addressed in the future. Some of them could be addressed in the long run, while others might shown some of the inherent limitation, and its harder to solve under current directions.

Reduced effects on cleaner transcription

Our approaches work best on noisy data and less so on cleaner data. This shows that our approach is limited to data that produces low ASR accuracy. As we noted in the Analysis section, if we perform rescoring on a perfect ASR transcription, it will harm the result if any rescoring happens. The next question that will arise after this observation involves determining the threshold of ASR quality to apply any rescoring algorithm. We believe that this is an important question, that suits research issues, since there could be different ASR performance thresholds for different rescoring algorithms. Our focus in this chapter has been to introduce the Word Burst phenomenon for recognition hypothesis rescoring, we did not a detailed comparison of different rescoring strategies. Hence, it is a possible direction for future work.

Recurrence of the same word

Our approach relies heavily on the recurrence of identical words. What if the word recurs in a different morphological form, or even recurs as synonyms? We did propose the target extension technique to address this problem, but we understand that this does not solve the core of the problem. The way we leverage Word Burst in this chapter is, "the same word is likely to cluster up within a small temporal window". What is the real communication unit in spoken language? We think the actual unit is semantic, not the words. The word is just a representation of semantics, hence using different words can still deliver the same communication unit (semantics). This thesis does not focus on semantics, since understanding semantics is another more challenging task. Still, we believe there are some questions that can be investigated by following this path. Identifying the relationship between different words is a good example. Our target expansion is not successful because we did not find a good model to represent the relationship between different words. Once we obtain a good way to represent and quantify the relationship between identical word and different words, the algorithm that leverage the identical words can be expanded to incorporate information from different words.

Singleton queries

Since our approach relies on the existence of multiple instances of the same word, when our target only appears once in the corpus, the improvement for those queries becomes very limited. Even with the well-known n-gram language model, people can challenge why it cannot address Out-of-vocabulary (OOV) well. We acknowledge this limitation of unable to process OOV words for our approach, and we think that exploring ways of addressing the instances that only show up once in the corpus will be an interesting future direction. Still, for this thesis, we will focus more on the target that follows our assumption: the same word that is likely to cluster up within a small temporal window. To go further on this directions, identifying whether singleton word demonstrate some special characteristic can help us predict or process singleton word in the corpus more efficiently. Or we can try to use other more frequent word to estimate the activity of the singleton word, and process according to the estimation. The fact that it only occurs once in the corpus indicate we should put focus not of the word itself but other words that could be related to it.

Parameter tuning on development data

Like other techniques, some training data is useful; the important bit is that there's a global phenomenon that can be leveraged, which we presented in this chapter. Our approach relies on parameter tuning using development data. This leaves our approach open to the criticism that it is not sufficiently robust because it requires tuning parameters by grid search for different data sets. Still, due to the limited amount of training data we have, its difficult to train statistical models, cause the training data we have could be biased and not able to represent the distribution of test data. Since we are not using statistical models, without even tuning the parameters of the development set, it will be an universal approach suitable for any data. It is difficult to solve problem without a task specific information. In the end, what we use is a general phenomenon tuned by task specific development data. In the STD task we have worked on in this chapter, the WER on each language is different, and the properties for each language and query are also different. The parameter tuning on development data is the necessary step to allow the algorithm to understand how it should process this specific set of data. If we are aiming to build model based on data, one possible way of doing that is to create a model from related domain that has more data available. Then build a way to utilize that well-trained model to our target domain, which might include transformation of trained model.

4.8.3 Future Work

Identify the real communication unit

As we discussed in the last section, identifying the most significant communication unit can be the real challenge, and we believe it is still an open problem. A key challenge in this direction is people has different interpretation of what semantic is, hence it's very difficult to standardize it. If we are able to identify the real semantics of the spoken word, when synonyms are present, the synonyms can be "normalized" into the same semantic token. In that case, when a computer

is processing human speech, it can have much better understanding and be more robust in recognizing what humans had spoken. This is because if we get the semantics right, the words are not that significant anymore. We think this is an interesting direction that can follow our work in this chapter. Once the semantic unit is explored and well defined, most of the language modeling work that focus on word tokens can be applied to those semantic unit as well. Also that should be able to affect many different language processing domain such as machine translation, since no matter what language people use, the goal for verbal communication is always delivering semantic information, and a generalized semantic unit should be able to represent all of them.

Beyond the identical communication unit

The Word Burst phenomenon described in this chapter works when we have identical communication units. However, the first part for our assumption is: "The information that is delivered in close proximity tends to be relevant to each other". We use this assumption to build up the relationship between the identical words, yet there is information that lies in the interactions between different words. Our unsuccessful attempt at target extension tried to address this property by using standard co-occurrence information such as mutual information and topic modeling. However, the Limited Resources Condition made it difficult to construct a robust model with a limited amount of data. Finding a way to model the relationship between different communication units without the need for a large amount of data for training is a possible future topic, as it can apply to many applications. This will also address one of the limitations of the work we presented in this chapter, singleton query. Even when a target word only occurs once in our entire testing corpus, its context communication unit can still provide us with much information about it. In order to move on in this direction, we think it might be necessary to have more relational information to tie up the relationship between words compared to today's n-gram language modeling. Most of the language modeling nowadays focuses on the occurrence of words and their relationship with the other words that are spoken around it, yet that definitely does not capture all the features that make people speak out a word. The context beyond the word token such as the information of the speakers should be all taken into consideration in order to achieve real understanding of communication units.

Integration into ASR operations

In this chapter, Word Burst had been used as a heuristic for hypothesis rescoring, or a feature in sequential labeling. It is also possible to use it in the standard ASR operational pipeline. A straight forward way of applying it is to use it similar to the way how cache-based language modeling is used, where the probably of the language models will be dynamic, and affected by the conversational phenomena that occur in the recognized text. When a specific content word shows up in recognized text, dynamically provide a boost (or penalty) on the probably of that content word in the language models. In addition, it is possible to have multiple language model prepared, and the ASR system can be switching between different language model to use according to the current conversation. The switch of the language model can be triggered by the occurrences of keyword, topic modeling, or even the user information (assuming the speaker information is available.) As long as there is a way to dynamically modify the model used by

the ASR system, leveraging conversational feature with the current ASR operation should be possible to achieve.

Using Neural Networks with Conversational Features

Neural Network based model have received a great deal of attention recently. We will discuss some of the potential approaches for integrating the conversational feature we introduced here into Neural Network based models. The simplest approach would be to create an extra feature dimension to represent the occurrence of specific conversational features. This will be similar to our approach for identifying recognition errors, where extra feature dimensions will be created by examining the raw data. For example, creating an extra dimension of features to represent whether the Word Burst occurs in the input or not. In addition to create extra features from examining the data, it is also possible to provide extra labels with conversational information to the input data. This direction includes labeling the conversational text with some extra conversation based labels (for example: beginning of conversation/ middle of conversation / end of conversation), and let neural network system be trained based on these extra conversational features. The key of using conversational information in neural network based model relies on finding a good way to represent the feature in the way neural network can digest and be trained from.

4.9 Summary

In this chapter, we described how to leverage the Word Burst phenomenon for hypothesis rescoring in Spoken Term Detection and for identifying recognition errors. We first discussed our motivation for leveraging Word Burst; since it is difficult to improve the performance on the ASR system, we decided to introduce structural knowledge about conversations to improve the recognition hypothesis quality. Word Burst turns out to be a good source of information, since it does not require large volumes of data to train and can be applied to multiple languages.

We then present two different rescoring algorithms, Word Burst rescoring and Unique Penalization rescoring. We also develop a target extension technique that can apply to Word Burst rescoring, which extends the effect to agglutinative languages. We also show that Word Burst has more general applications, such as a feature to identify recognition errors. For STD, our experiments were conducted on five different languages, and our findings show that our approaches improved the STD performance, mostly on false alarm reduction. We then provide analysis of different phenomena we observed in our experiments, and discuss the limitations of our approach. For identification of recognition errors, we conducted experiments on three different languages and different WER. The presented work concludes that using the Word Burst phenomenon can be beneficial to STD under the Limited Resources Condition and identifying recognition errors. These will eventually lead to better extraction of knowledge from spoken data.

Chapter 5

Integration of Different Recognition Hypotheses in Spoken Term Detection

5.1 Motivation

Aside from improving the quality of an Automatic Speech Recognition (ASR) system or using Word Burst rescoring as we proposed in the previous chapter, the other way for improving Spoken Term Detection (STD) performance under Limited Resources is to identify whether there are useful features from the existing systems that have not been used (Chiu et al., 2014). The standard pipeline for an STD system can be considered as a way of communicating information through different system components. For example, lattices or confusion networks can both being used as decoder output. Is there any valuable information that is unique within each specific structure? Does each structure has its own specific error pattern, that we can reduce the error by leveraging other structure that has different error pattern? Can integrating different structure benefit us, despite the fact that the ASR system is completely identical? We aim to investigate whether there are unique information lies within each structure, and how can we leverage those to achieve better STD performance. This research provides new direction for analysis on the strength and weakness of each individual systems and new approach to combine different systems to achieve best end results.

5.2 Our approach

Figure 5.1 shows where our approach locates within a standard STD pipeline. We use an ASR system to output lattices/confusion networks (introduced in Section 3.1.3) as decoder output. Instead of rescoring the hypothesis as in the last chapter, we perform term detection on each of the decoder output individually, then introduce a decision process that performs system combination on two different detection results. Both lattice and confusion networks have their own specific error patterns, so our goal here is to integrate both structures to obtain results that have less errors compare to each structure individually. In this way, we can achieve better STD performance with the same ASR system. The assumptions in our approach are as follows:

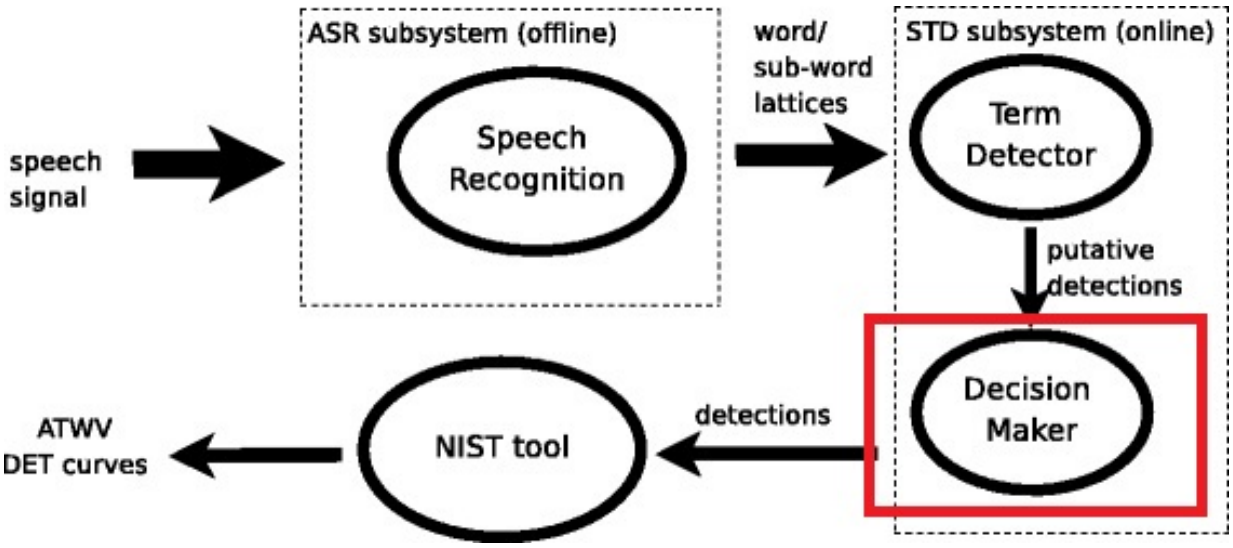

Figure 5.1: The components on which we focus in the standard STD pipeline for this chapter

- When the same information is communicated in different structures, there is unique information that contained in each structure.
- Consequently, integrating the information from different structures delivered from the same system can provide better understanding of the original data.

The intuition for leveraging this mismatch between the recognition hypothesis is as follows. When people are communicating information to various audiences, the way they structure their information will be different. When communicating with a person who has broad domain knowledge, the information can immediately go into detail and skip the unnecessary background knowledge. On the other hand, for a person who is very unfamiliar with the field, we might want to focus more on the general concepts and skip some details.

Considering the same situation, but now replacing the entire processwith an STD system, the speech corpus we want to search is the original information, and the ASR system is the person who knows this information. When delivering the original information to different people (in the STD case, decoding the speech corpus into different decoder output), the delivered result has a different structure. When presenting the recognition hypothesis as a lattice, it focuses more on preserving context and historical information; this can be considered as one of the approximations of the original information. On the other hand, when presenting as a confusion network, its focus becomes representing the recognized result in a compact and aligned way, which is another kind of approximation. There could be unique information available within each of these approximations of the original information, because each structure has different error patterns, as well as the correct patterns. Combining them can give us a more clear picture of the original information compared with any of the individual sources of information. The key observation is that there is information separate from the original generation of the information.

5.3 Search and Combination Description

There are two search methods we are going to use for different structure of recognition hypotheses. We aim to achieve improvement with the combination of the two different search results.

5.3.1 Finite-State Transducer (FST) Search

The lattice was first introduced in section 3.1.3 and can be considered as a FST. Lattices are probability networks of the possible decoder output. A lattice contains a set of word hypotheses with boundary times and transitions between different hypotheses (Ortmanns et al., 1997). It can be found that a lattice tends to contain a large number of word hypotheses including both the true hypotheses and the competing hypotheses. The 1-best decoding hypothesis can be created by following the most probable path in the lattice.

FST search is conducted on the lattice generated from an ASR system. A more detailed introduction of lattices is presented in Chapter 3. Since a lattice is a recognition hypothesis that contains better context information and more information, searching through a complicated lattice is considered more complicated. Our FST search pipeline is described in (Chen et al., 2013a,b) is capable of both in vocabulary (IV) and out of vocabulary (OOV) search. We implement the lattice indexing algorithm proposed in (Can and Saraclar, 2011) making use of the Kaldi toolkit (Povey et al., 2011). The search is separated into two steps, indexing and search.

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 define at the word level, which is an essential part of our lattice generation procedure (Povey et al., 2012). Otherwise, the indexing algorithm tends to blow up since the number of potential word sequences grows exponentially with the sequence length, if we use sub-word level unit such as phonemes to build up the lattice.

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) (Chen et al., 2013b) 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 performed 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.

5.3.2 Confusion Network(CN) Search

CN search is conducted on the confusion network, which is also introduced in Chapter 3. A confusion network is a more compact recognition hypothesis. As a result, searching through a confusion network is much simpler. Still, we believe there could be unique information that can be discovered in this compact recognition hypothesis, such as the extra link between word hypothesis that are created when building the confusion networks.

Our procedure for generating confusion networks is based on the Minimum Bayes Risk decoding algorithm of (Xu et al., 2011). 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, which means no hypothesis during the period of time of that path. Each path generates a different detection: the starting and ending times are those of the first and last clusters in the path, respectively, 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.

5.3.3 Algorithm Development Process

Different hypothesis representation were proposed as the format for ASR output. (Ortmanns et al., 1997) proposed using lattice as the structure for recognition hypothesis, while (Mangu et al., 1999) suggest confusion network as an alternative structure. The research following these two structures both achieved successes, including (Miller et al., 2007) and (Mamou et al., 2007). However, we did not find any work trying to compare two different approach directly. So our first attempt is to do straight up comparison between two representations. After noticing that the FST and CN searches have different strengths, we start to investigate approaches to leverage this feature on mismatch. System combination as a popular way in STD community becomes the way we choose to leverage this feature (Mamou et al., 2013).

The main parameter tuning that can happen in this experiment is the weight for the different systems that we used to combine. However, we decided not to do any tuning on this parameter, since the extra gain from tuning the weight of different systems does not contribute to the point we want to make, which is each representation should have some unique information and we could leverage them by combining different representations. The weight for the system that are used for combination are always equally distributed.

5.3.4 Search Combination Techniques

After we obtained the detection result from two different search methods, we perform system combination on it. Search results from different search methods are combined on a per-keyword basis. For each keyword, its detections in all of 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 reason for this approach is to create every possible detection according to the pre-combined result. 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 (Mamou et al., 2013):

- *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. These three methods showed most promising result in (Mamou et al., 2013), which is why we decided to use them.

5.3.5 Difference between current approach and previous work

System combination is not a new concept in the STD task. See section 3.2.2 discussing the work on system combination in Limited Resource Condition STD. The main difference between our work and the previous work is that the previous work only focused on combining the results from multiple ASR systems (Mamou et al., 2013; Mangu et al., 2013), while our combination is based on the result from a single ASR system. With our approach, we are trying to explore the full potential for a single ASR system. The insight we wish to provide is that, even with the same ASR system, by leveraging the differences in decoder output, we can still yield additional improvement. Aside from using single ASR system comparing with multiple ASR systems, our approach can be considered as doing system combination in the different stages of the STD process, and that is a hitherto unexplored location to do so. Previous work (Karakos et al., 2013; Mamou et al., 2013; Mangu et al., 2013) perform system combination at the very end of the processing pipeline, when the STD detection result from different ASR systems are created. Our approach perform the combination right after the ASR is finished, but the decoder output is stored in different formats. This enable us to do combination on the different decoder output from the same ASR system. In the previous work, the gain from the combination comes from different ASR system configuration, while in our approach, it comes from different structure of the recognition hypothesis. This is interesting becaues it showed that even training with identical training data, if the result are in different type of representations, it still contains unique information that can be used.

5.4 Dataset and experimental setup

We describe the details of our experiments in this section, including the dataset we use in our experiments, the experimental setup, the evaluation metrics and the tool we used.

5.4.1 Dataset

We use five different datasets to test/assess the generality for our approach. Those are conversational (telephone) speech recorded in five different languages: Assamese, Bengali, Haitian, Lao, and Zulu, as available in the IARPA BABEL program (Karakos et al., 2013; Mamou et al., 2013). 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 Experimental setup

STD system description

Our STD system uses an ASR and term searching two-stage pipeline, which is based on the Kaldi[1] toolkit (Povey et al., 2011). The decoded hypotheses are represented as lattices, and then converted to confusion networks. Each search applies to the respective hypotheses representation, and the detection results are combined with different search combination techniques.

The following figure, Figure 5.2, shows the search combination pipeline that was used in our experiments.

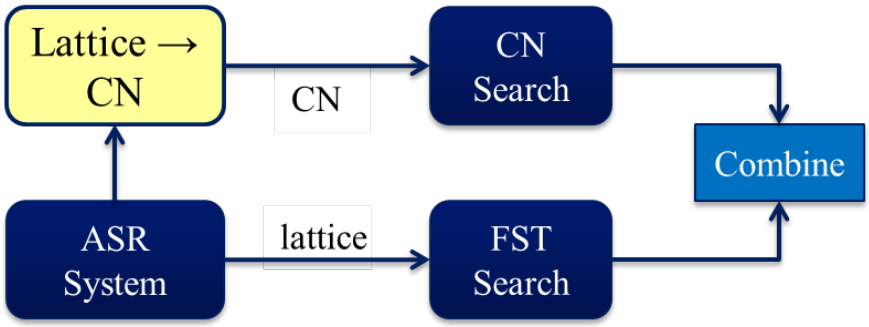

Figure 5.2: Search combination pipeline

Evaluation Metrics

The evaluation metrics for STD were introduced in section 1.3.1. The formula for Term Weighted Value (TWV) is as follows:

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

In this chapter, we use two additional separate metrics based on TWV to describe the performance of STD systems:

- Maximum Term Weighted Value (MTWV): MTWV is the maximum TWV computed over the range of all possible values of the detection threshold. It estimates the performance for the detection result ranking list without considering the right 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 more information for the overall quality of our search results, as they are not sensitive to specific detection threshold. With the appropriate detection threshold, the ATWV (the metric used in the last chapter) is very close to the MTWV value, because that

[1] http://kaldi-asr.org/

means the threshold for ATWV is been set close to the optimal place for ATWV, and MTWV is result you get when you have the exact optimal threshold.

Because we are using different evaluation metrics, we think it will be difficult to express the performance of our system with existing IR metrics (unlike chapter 4.) The reason of that is, in chapter 4, we compute ATWV, which means we have a threshold for deciding hit and miss. With the threshold being clear, it is also easy to compute the precision and recall. MTWV and STWV are two metrics that only focus on the order of the result, hence the threshold is not available and it is difficult to compute Precision, Recall and F-score. As a result, in this chapter we will focus on reporting the TWV-based metric.

Code

We published the code that had been used to completed the experiments on github [2]. Note that the code in this repository requires the data and scoring script coming from IARPA BABEL program to make it work, which must be obtained separately. But we still provide the code that conduct the system combination, which we think can demonstrate the algorithm we used in experiments. The code combines two or more different search results, and by feeding in search result creating with different system configuration, we can preform the experiments reported this chapter.

Description of Experiments

We carried out three different sets of experiments. Each set was conducted on three different decoding front ends: 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, since the query turn will not be presented in recognition hypotheses.

The first set of experiments compares the performance of the two different searches, FST search and CN search. The second set of experiments combines the search results from FST search and CN search to determine if we can obtain better STD performance. The final set of experiments combines all of our results to determine whether the gain from the individual systems is additive. The combination is also performed in different orders to note whether this affects the final result. We did not expect that final result will change yet still provide these result for completeness.

5.5 Experimental results

5.5.1 Comparison between FST and CN Searches

Figure 5.3 shows the MTWV for different system configurations on five different languages (Assamese, Bengali, Haitian, Lao, and Zulu) and 3 different decoder front-ends: DNN, BNF, and PLP. We performed a statistical analysis by fitting a general linear model to the data and found statistically significant differences between languages, front-end features, and search methods,

[2] https://github.com/jltchiu/Combination

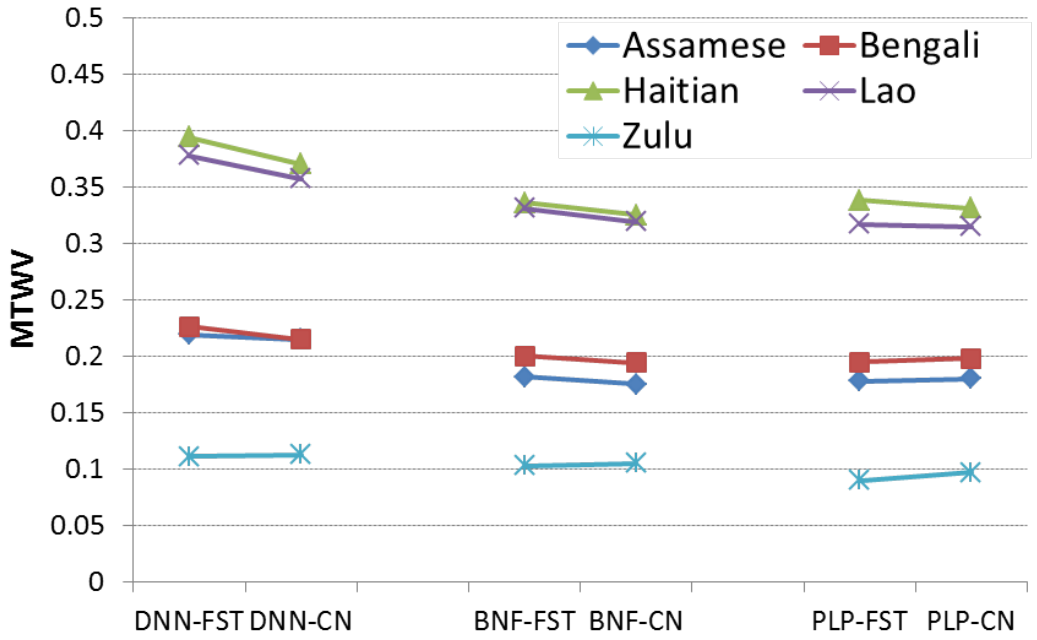

Figure 5.3: System comparison between different ASR systems and search methods.

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, as we will explain in section 5.6.1.

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; we believe that this is because it considered the score for both inputs, which is better than the CombMax approach; it also gives uniform weightings from both system, which is better than CombMNZ. The results shown in Table 5.1 are averaged over front-ends. 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 5.3), which we will discuss in depth in section 5.6.1. Second, CN search has better performance on Supreme Term Weighted Value (STWV) over FST search. This is caused by the conversion from lattice to CN. The details for both observations are discussed in the Analysis section.

5.5.3 Combination between decoding systems

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

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 search 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

After combining the results from multiple searches, these results are further combined with the results from different decoding systems to achieve even greater improvement, as is shown in Table 5.2. The result is the average Maximum Term Weighted Value (MTWV) over all languages. We select the DNN system as our single-best system. By search combination, we achieve better performance on all five languages. If we combine the search combination results 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 with the DNN system, combining them nevertheless yields improvement. We also tested doing system combinations in different orders but found that the order of combination does not have much impact on performance. The improvement from combining different decoding system have multiple causes. By using different training data or features, each decoding systems will have its own unique error patterns, and combining the result from multiple systems can create a result that utilize the strength of each systems, which is usually beneficial as reported in previous works (Karakos et al., 2013; Mamou et al., 2013; Mangu et al., 2013).

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. Upon closer inspection, we found that the difference is due to the distributions of query length for each language. Each of the 5 languages has around 2,000 queries, yet query length is distributed 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
Total	1959	1970	1924	1925	1985

The queries for Haitian and Lao have relatively low percentages of queries with length 1. On the other hand, Zulu has extremely high percentage of queries with length 1. This distribution is highly correlated with the result shown in Table 5.1, where it shows that 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. Unlike the analysis we presented in last chapter which focuses on the number of occurrences for the query word, the analysis here focuses more on the length of query, and the occurrence is not the main focus here.

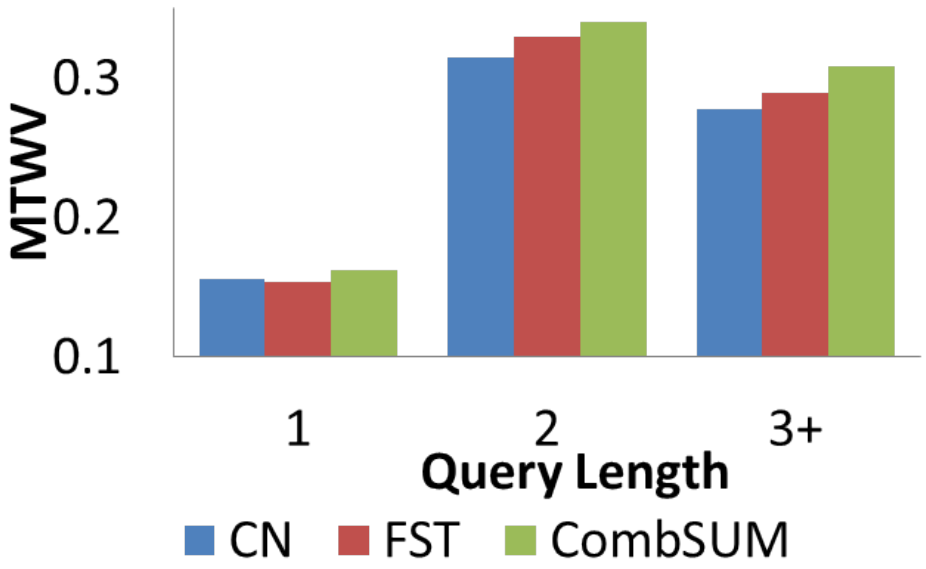

Figure 5.4: MTWV interactions for search methods and query length

Figure 5.4 shows the interactions between search methods and the query length, averaged over all languages and decoding systems. This analysis yields two findings.

First, CN search performs somewhat better on queries of length 1 word, while FST search outperforms CN search on longer queries. Also, CN search has fewer false alarms compared with FST search on the 1-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 a better MTWV score. FST search outperforms CN search on multi-word queries. This is because lattices can better preserve history information for decoding hypotheses compared with CN. This observation provides an explanation for the result shown in Figure 5.2, where FST search outperforms CN search on every language except for Zulu. From Table 5.3, 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. We believe the reason why Zulu query are mostly single word query is because Zulu is an agglutinative language, and morphological variant of a word can already express rich meaning. In this case, there is less need for having multi-word queries.

Second, search combination provides better performance on multi-word queries, compared with single word queries. This matches our finding in section 5.5.2 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 little difference between the 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. The implication for system design on this is, the performance gain from combination came from the part that the data is represented differently. Combining multiple system that have similar representation does not have significant gain on performances.

5.6.2 Search and ASR systems

Figure 5.5 shows the interactions between different ASR systems and search methods. The result is the average MTWV over all languages and two different search methods. We have two observations according to this analysis. First, search combination provides consistent improvements across different decoding systems. This indicates that 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 searches 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

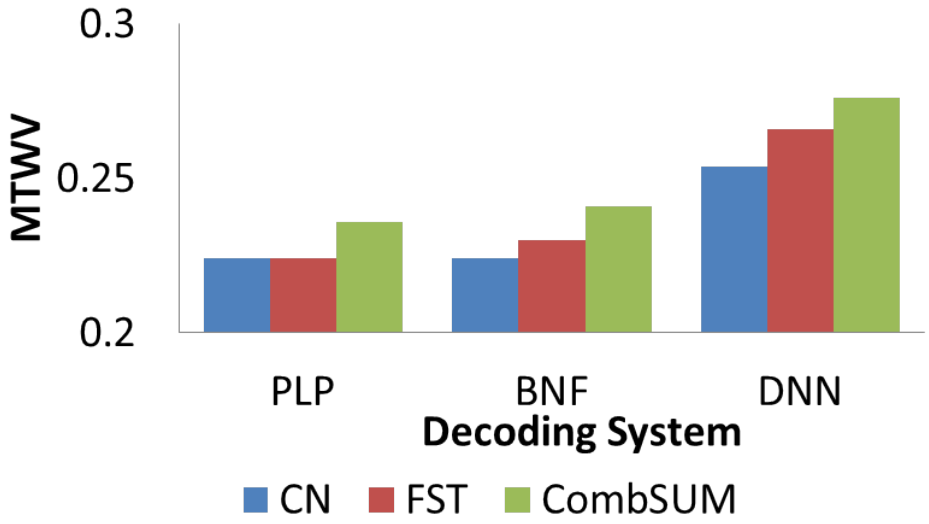

Figure 5.5: MTWV interactions for search methods and ASR systems

can still achieve extra improvement.

5.6.3 The higher Supreme Term Weighted Value (STWV) in CN search

From Table 5.1, we can see that CN search consistently has higher Supreme Term Weighted Value (STWV) compared with the FST search. This is because the creation of confusion networks gives rise to additional 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 5.6 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 significant impact on the MTWV score. FST search still produces a better MTWV score over multi-word queries.

5.7 Discussion

5.7.1 Contribution from this Chapter

The general contributions we made in this chapter include:

- We identify a phenomenon that delivering the original information through an ASR system with different decoder output leads to different structure, and there could be unique feature that has been stored in the mismatch of each structure. (5.5.1)

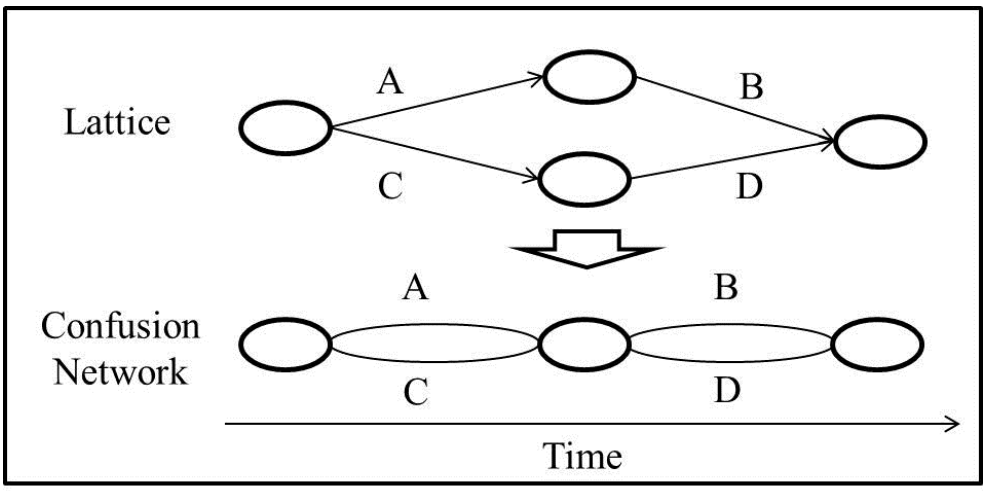

Figure 5.6: Extra link created during CN conversion

- We also validate our assumption that combining the original information that is delivered by same ASR system in different structures can give us a better picture of the original information. (5.5.2)
- The improvement mostly comes from the difference between how the target processes the data provided by the recognition hypothesis with the different structure. (5.5.2)
- We verify that our assumptions can be applied to multiple languages, which means that it is not a language-dependent phenomenon. (5.5.2)
- We also showed that it works on different front end of the ASR systems for identical input speech. (5.6.2)
- Even with multiple ASR systems, each ASR system can benefit from the technique we described in this chapter, and combining multiple ASR systems still achieve even better performance. (5.5.3)

The task-specific contributions we made in this chapter are:

- We designed a search system combination framework based on a single ASR system that improves STD under Limited Resources Conditions for multiple languages. (5.4.2)
- Our approach consistently improves the STD result on different ASR systems, which means that it can work on different qualities of ASR systems. (5.5.2)
- We also found that combining different ASR systems after we perform our search combination achieves even better results, making it an orthogonal improvement for the common approach. (5.5.3)
- Our combination contributes most on multi-word query, on which the search strategy for FST search and CN search has the most difference. It shows using different form of decoder output from same ASR system can still yield improvement. (5.6.1)
- The confusion network generation process could create a small amount of additional multi-word path, which can improve the overall recall of the recognition hypothesis. (5.6.3)

- We identify the performance gain from combination came from the part that the data is represented differently. Combining multiple system that have similar representation does not have significant gain on performances. (5.6.1)

5.7.2 Unresolved Issues

Single-word query

As we discussed in the Analysis section, the improvement due to our approach mostly comes from multi-word query. For single-word query, it does not provide significant gain. The reason for this is because both search approaches process single-word queries in a very similar way, so there is not much difference between the two searches, and less unique information for each search can be discovered through combination. This means that, for our approach to work, the key is to combine different ways of saving/using information. A possible follow-up question for this observation is whether it is possible to explore a systematic method for identifying different approaches to process any task that can be used to combine and gain improvement like we described in this chapter. Since the method of processing the information depends on each individual task, and this thesis is focused on leveraging the difference in decoder output rather than building a systematic combination pipeline, we believe this topic is a subject for future research.

Since our approach primarily works on multi-word queries, we should also compare with it some Spoken Document Retrieval (SDR) approaches. There are many fundamental differences between the two tasks. First, the size of multi-word query an STD task is about two to three words, so that in total the time length of the search target will be less than 5 seconds, while the smallest size for SDR retrieval is around 30 seconds, which is much longer. Most SDR approaches require Retrieval Models (Chiu and Rudnicky, 2014), that are based on a significant amount of context that is part of a larger document, while the STD does not address the context at all. Second, STD focuses on matching the exact query, while SDR aims to find relevant segments, which does not require exact matching. Due to these differences, we believe that these approaches are not comparable.

Combining with more systems

In the experimental investigation section, we showed that the gain we observed from our search combination can improve with the standard ASR combination. This means that even if we can extract more information from a single ASR system, adding more ASR systems is still helpful for identifying the original information. However, when there are far more ASR systems available, does our approach, which can extract more information from a single ASR system, become necessary? We think that when combining with additional ASR systems, the gain on the single system with our approach could only provide marginal effect on the final multi-ASR system combination result. We do our search combination based on the combined ASR result from 3 ASR systems and the performance gain is not significant:

This is because the most important gain from our approach comes from the difference between different structure of decoder output. For example, how lattice and confusion network handle multi-word query differently. The more differences there are, the more improvement we

Table 5.4: MTWV/STWV from search combination based on combined hypothesis

Language	Metric	Combined FST Search	Combined CN Search	Search combination from both
Assamese	MTWV	0.248	0.228	0.249
	STWV	0.448	0.457	0.466
Bengali	MTWV	0.256	0.238	0.257
	STWV	0.427	0.437	0.445
Haitian	MTWV	0.420	0.391	0.422
	STWV	0.577	0.584	0.598
Lao	MTWV	0.413	0.387	0.418
	STWV	0.562	0.567	0.584
Zulu	MTWV	0.125	0.124	0.126
	STWV	0.276	0.279	0.279

can achieve. However, the difference will be more significant if it is from different ASR systems, cause it could have the difference from different training data or feature, and those will be the key for improvement during combination. The original speech will have significantly better approximation from the information provided by different ASR systems than from the different decoder output provided by the same ASR system. Hence, we think it could be less effective under that situation.

5.7.3 Future Work

Smart integration strategy

In this chapter, we have worked on the combination of STD results based on different decoder output. The combination technique we use considered that both systems are equally important, which is probably a non-optimal solution. For example, in our experimental setup, the FST search has better performance on multi-word queries. With information like this, we can design a better integration strategy according to the property of the query. For a new or unknown system, it is also possible to design a task-specific or system-specific combination technique, which are not addressed in this chapter. We focus more on "what" to combine rather than "how" to combine, yet there is definitely a space for research on this problem, such as optimizing the weight of different systems with certain defined objective functions.

Integration beyond Spoken Language

In this chapter, our integration has focused on spoken language, yet there are still different forms of communication that can be explored and potentially integrated to better understand of the original information. Is it possible to integrate the structure between the words spoken by a person and his action or thoughts? For example, guessing emotion from the text a person generate? (Alm et al., 2005) This topic bridges the speech community with other research communities, and many research questions suggest themselves. Moreover, will the structure of spoken language for a person be different when they are in a different emotion or mind state such as lying?

Can a specific hand posture or shake of the head be correlated with certain words that are spoken? We think that cross referencing/integrating different forms of communication can give us a clearer picture of how human communication works. For example, when visual input and spoken input are both available, the object that are seen in the visual input can be used to provide topical information for support processing the spoken input.

5.8 Summary

In this chapter, we investigated the error patterns that we observed in different decoder outputs, and use combinations to improve Spoken Term Detection. We first discussed the motivation for our approach, if an ASR system can create different decoder output, each format of output might have unique error/correct pattern. If we can capture the unique correct pattern in each of the decoder output, we can use them together by combination. We designed a STD search combination technique based on this assumption, and determined whether it can provide consistent improvement on the STD result. Our experiments were conducted on five different languages, and our findings show that our approach improved STD results, mostly on multi-word query since the search was processed differently in each individual system. We present an analysis covering multiple perspectives, including how query and ASR systems interact with different ways of system combination. The work in this chapter concludes that leveraging the mismatch of the information delivered and processed in different structures can achieve better understanding of the original data, even if the ASR system is identical.

Chapter 6

Distributed Representation of Utterances

6.1 Motivation

In previous chapters, we address Spoken Term Detection (STD) problems with different approaches. However, there are fundamental problems for the STD task that still have not yet been addressed. Even if the query term can be detected perfectly, the user still might be confused by a multi-sense word. Detecting the same word does not guarantee that the user can find the content he or she really needs, since the sense for query term that is not desired by the user is still not useful. For example, when the word "bank" are detected in STD system, showing the user who is looking for "river bank" the location of the word "financial bank" might not be useful, even though they shared the same written form. As a result, in this chapter, we investigate a new tasl, to the Spoken Word Sense Induction (SWSI) (Chiu et al., 2015). The goal of SWSI is to identify which sense of a multi-sense word is used given the context of the spoken word and to allow the user to select detections according to their needs. The difference between "induction" and "disambiguation" is that induction task requires the condition of not using any external information/knowledge. We construct the distributed representation of utterances by leveraging the word and its context. The distributed representation turns the word in a large corpus into a point in a high-dimension vector space, and the distance between different points in the space represents the relationships in meaning between different words. We aim to improve our understanding of query terms with this model.

6.2 Our approach

Figure 6.1 shows where our approach fits within a standard Word Sense Induction (WSI) pipeline. Note that for SWSI, it is just necessary to replace the text document in the figure with the ASR result. We leverage our knowledge of the relationship between word and its context to create a better feature space to represent every spoken utterance, since we believe that good features are the key to success in the SWSI task. We constructed our features based on two assumptions:

- The spoken words that show up with similar context should have similar meaning relative to each other

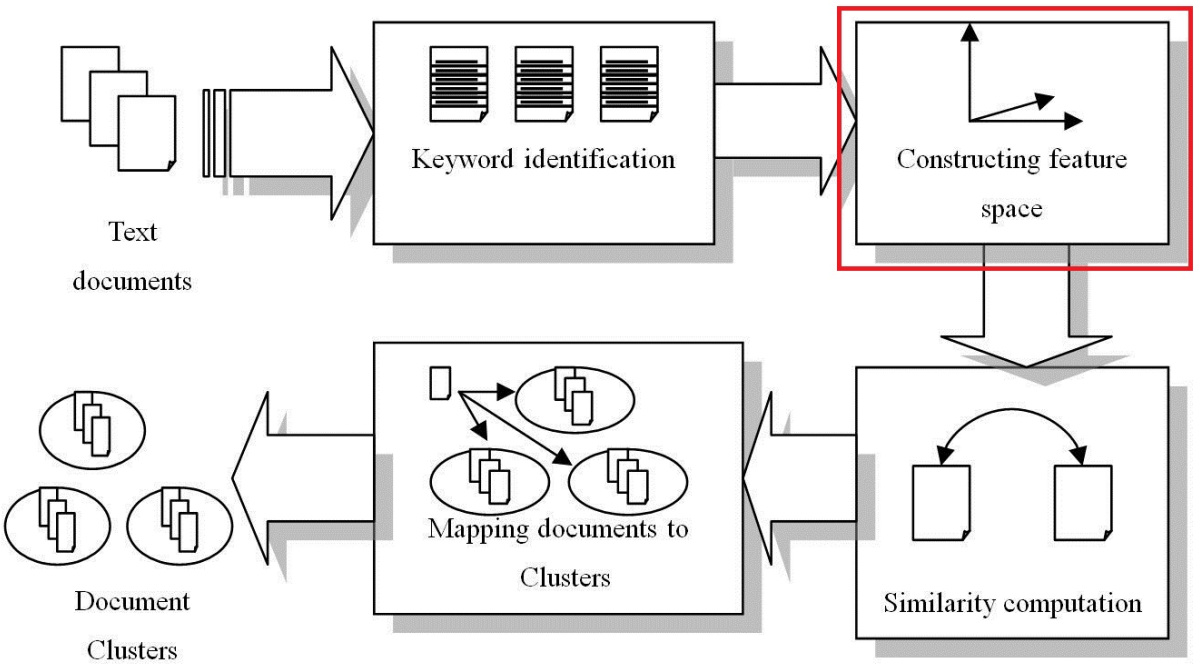

Figure 6.1: The components on which we focused in a standard WSI pipeline for this chapter

- If the spoken words occur in very different contexts, they are less likely to share the same meaning even if the words are the same orthographically

We carefully avoid the use of the word "semantic", since we believe that this might be a statement that is too strong. We are simply modeling the similarity (how likely two different words show up in similar context) of the word in the SWSI task, and we aim to separate different words' semantics according to their similarity. The reason why we said that word is only an approximation unit is because we believe the actual unit of communication is semantic, as discussed in Chapter 4. We leverage the relationship between the word and its context to create a vector space (word embedding) that can represent the relationship between different words. Within the word embedding, the distance between different words reflects how likely they show up in a similar context. This vector space can help us to model the relationship between words. However, a word can show up in very different context if it has multiple meanings, and this phenomenon can not be represented well in the current word embedding, as every word is represented as a single point in the space. As a result, we created another point in the space that represents the word itself together with the utterance it belongs to. These utterance points can be used to identify the meaning of difference instances of the identical word.

Our word embedding is trained on the ASR transcription of the data we want to perform SWSI. This is significantly less data than typically way for training embedding. We picked 100 as the number of dimension for our word embedding training, which is the same with the best parameter reported in (Dai et al., 2015). We did not focus on tuning between different numbers of dimensions for word embedding, as our focus is on tuning the number of clusters and comparing with different algorithms. Also, since we want to identify approach that can

potentially be applied on the language without much data available, we did not explore the usage of pre-trained embedding, as high quality embedding might not be available in every languages. The positive result for our approach suggest it can be port to languages with insufficient data available.

6.3 Extracting Features with Word Embedding

6.3.1 The Skip-gram Model

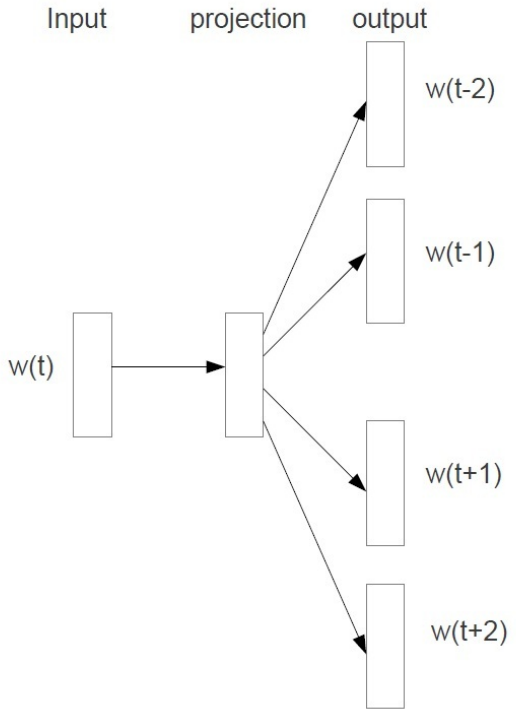

Figure 6.2: The architecture for Skip-gram model (reported in (Mikolov et al., 2013))

(Mikolov et al., 2013) recently introduced the Skip-gram model, and the illustrative architecture for the model is presented in Figure 6.2. Skip-gram models and other Neural Network Language Models (NNLM) produce representations for each word in the training data according to its surrounding words. Each word can be viewed as a point in a "Word Embedding" space, and if there are two words that are located closely in this space, it means that those two words tend to show up in similar surrounding word contexts in training data. The advantage of using the skip-gram model instead of other NNLM is that the skip-gram model requires much less computational resources yet it can still achieve good performance. The formal definition of the skip-gram model is as follows:

Given a sequence of training words $w1$, $w2$, , wT, the objective of the skip-gram model is to

maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}\log p(w_{t+j}|w_t)$$

where c is the size of training context (which is a function of the center word w_t). Larger c results in more training examples and is more likely to achieve higher accuracy, yet it requires more training time. The basic skip-gram formulation defines $p(w_{t+j}|w_t)$ as

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W}\exp({v'_{w_O}}^{\top} v_{w_I})}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary.

This formula is difficult to use in practical applications since the cost of computing increases with W, which is usually significant (i.e. the size of vocabulary). Instead, we use Negative sampling (Mikolov et al., 2013) to replace the $\log p(w_{t+j}|w_t)$ in the original Skip-gram objective. The idea of Negative Sampling is to only update the weight of a small number of negative words, instead of every negative words in the vocabulary during each training iteration, which can significantly speed up the training process. The Negative sampling is defined by:

$$\log\sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} E_{w_i\ P_n(w)}[\log\sigma({v'_{w_O}}^{\top} v_{w_I})]$$

The task is to distinguish target word w_O from draws from the noise distribution $P_n(w)$ using logistic regression with k negative samples for each data sample. We use k=5 in our experiments.

Another factor that will affect our training is the frequent words in our training data. According to Zipf's law, the frequency of any word is inversely proportional to its rank in the frequency table. This phenomenon causes frequent words to provide less information compared with rare words. We perform a simple subsampling to counter this imbalance in word frequency. Each word w_i in the training data is discarded with probability computed by:

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

$f(w_i)$ is the frequency of word w_i and t is a chosen threshold. In our experiments, we set t as 10^{-4}. This formula was chosen because although it subsamples words fairly aggressively, it preserves a word's rank according to its frequency. Others (Mikolov et al., 2013) have reported that this approach significantly improves the accuracy of learned vectors for rarer words.

The Skip-gram model will produce a single point in the "Word Embedding" space for each word in the training data. However, this is actually a limitation of the model, as each word is forced to be represented as a single point. This is not an ideal situation, because if the w has different meanings, it is likely to occur with very different surrounding words. The computed single point for w is the average of all instances of w, which conflates the different meanings. If

sense-labeled training data are available, then it would be possible to train multiple distributed representations that differentiate the different meanings of the same word, yet such data would not be available in a typical SWSI scenario, however.

Algorithm Development Process

As noted in the last section, the limitation of the existing Distributed Representation based model is that each word is only represented as a single point, and that is not helpful for addressing our task. However, in an usual WSI setup, the way to separate identical words is to leverage the other word around it. We believe the same strategy can be applied to Distributed Representations, so we set about to develop approach that can achieve this effect, by creating multiple points in the Word Embedding space from more than one words. There is preliminary work from (Mikolov, 2012) which mentioned a way of creating Distributed Representations for units that is bigger than word, mostly for documents and paragraph. We decided to adopt their technique for our tasks.

The main parameter that requires tuning for our approach is the number of clusters. We understand that this is a challenging parameter to decide, so we reported the result on the parameter we had tried. Deciding what is the optimal number to use is still an open question, so we provide result on all potential settings. This parameter will be discussed more in the analysis section.

6.3.2 Distributed Representation of Utterances

In order to overcome the limitation of existing Skip-gram models, we use a distributed representation for utterances to differentiate the meaning of multiple instances of the same word. Our intuition is that if we can obtain the distributed representation for the entire utterance, which contains our target word and the surrounding words, we can then use that representation to differentiate the meanings of a specific word. Thus, if the meaning of the utterance is different, we can expect that even a different instance of the same word in an utterance is likely to have a different sense. The SWSI task is usually considered to be a clustering task; clustering the utterance instances can be a good approximation of clustering the words by sense.

We obtain the distributed representation of an utterance with the following setup: We assume there is an extra "utterance token" at the beginning of each utterance. This token will be trained with every other word in the sentence. So given a sequence of training words w_1, w_2, ..., w_T in a specific utterance, the objective of the distributed representation of the utterance is to maximize the average log probability

$$\sum_{t=1}^{N} \log p(w_t|u)$$

where N is the size of the entire utterance and u is the "utterance token". This will map the utterance into the same space with other words in the training data. The output of this representation has multiple applications, it can be used as a representation of utterances for clustering purpose like what we do here, or the cosine distance of different utterances can be computed to

provide the relative relationship between different utterance, which gives us a better understanding on the semantic level. Having utterances and word represented in the same space indicates that we can use different number of words to communicate a specific semantic, which is true in human conversations.

The intuition of this approach is to create new points in the vector space depending on the whole utterance that our target word shows up. These new points models how different instance of same word can show up in different context. In this way we are able to model a multi-sense word with multiple points, instead of a single point in the vector space.

6.3.3 Difference between our approaches and previous work

Much previous work has been performed on the distributed representation of words (Bengio et al., 2003; Hinton, 1984) as we discussed in Chapter 2. The purpose of all of these works is to project words into a word embedding space in which the relative distance between words in the space can represent the relationship between words (Mikolov, 2012). Our work also follows this strategy to create the word embedding space, yet we focus on separating multiple instances of a single word according to sense without any external information, which has not previously been addressed. Most of the earlier word embedding applications assumes that each point in the word embedding space can represent a specific word (Elman, 1990), yet in reality a multi-sense word should have multiple points in the space to represent its most appropriate location. Our work in this chapter leverages the concept of the utterance vector to separate multiple instances of the identical words.

Another major difference between our work and the previous work is that most of the earlier work trained the word embedding space with a very large corpus like the Wikipedia dump (Mikolov, 2012; Turney et al., 2010). In our experiments, we trained the word embedding space on the data that has been decoded by the ASR system. This means that our word embedding space will have much less data to work with, yet it will not have any mismatch with the data we are processing. Without a large corpus like Wikipedia to provide general contextual information for words, it is possible that some multi-sense words will be considered as single sense, because the there is not an example for every sense of that word in the data we are processing. We believe this is a strength and a weakness at the same time. The upside of our setup is that we will not have irrelevant common background knowledge that affects our performance, so our system will not have a strong tendency for the most common sense. However, the bad part is that our space will be less robust and if the data are not sufficient, we cannot create a robust space for our approach to work.

6.4 Experiments

We want to validate our hypothesis: " Distributed Representation of utterances is a good feature to separate the meaning of multi-sense word" with our experiments. This can be achieved by comparing what we propose with approaches that had been reported before. We will introduce the detail for our experiments in this section, including the dataset we use in our experiments, the evaluation metrics and the tool we used, and the experimental setup.

6.4.1 Dataset

We use 60 hours of YouTube "How To" video for our experiments. The reason we selected YouTube video as our data is because this is a real-world data set, and the video we used as data also has user-uploaded subtitles, which can be used as the reference transcription. The YouTube video corpus (Yu et al., 2014) includes human transcription, allowing us to compute the WER for ASR.

The ASR system we use to decode the speech is based on the Kaldi (Povey et al., 2011) toolkit. We have two different setups of acoustic model training to simulate different WER, which were 39.13% and 19.95% (nominally, 40% and 20%). Note that we do not intentionally create model that will generate 20% and 40% WER, it just happens with the different training data we use to create the acoustic model. The acoustic model of the 40% WER system is trained on the Wall Street Journal corpus consisting of approximately 80 hours of broadcast news speech. The 20% WER system's acoustic model is trained on 360 hours of video data that are in the same domain as the testing data. Speaker adaptive training (SAT) is conducted via feature-space MLLR (fMLLR) on LDA+MLLT features. DNN inputs include spliced fMLLR features. All decoding runs use a trigram language model that is trained from 480 hours of YouTube transcripts (Yu et al., 2014). The 40% WER system is meant to simulate a mismatch between training and testing data, common in real-world use cases; it is about the same level as reported in (Liao et al., 2013). The 20% WER system represents a more controlled environment (or more accurate ASR), as the mismatch between training data and testing data is much smaller. Together with the human transcription that is nominally 0% WER, we expect that this can provide insight on how ASR performance affects SWSI performance. The number of word tokens and the vocabulary size are reported in the following table:

Table 6.1: Vocabulary size and number of tokens.

WER(%)	**40**	**20**	**0**
Vocabulary Size	55266	52377	55162
Number of tokens	715849	745402	742260

To select the target queries for our SWSI task, we adopted the query selection process used in the SemEval-2013 WSI task (Navigli and Vannella, 2013). We selected those queries for which a sense inventory exists as a disambiguation page in the English Wikipedia[1]. Additionally, the queries we selected each have 3 senses among the WordNet 5,000 most common senses (Clark et al., 2008) to ensure that the difficulties are comparable. Every query appears at least once in our 60 hours of YouTube data. In total we selected 125 queries.

6.4.2 Evaluation Metrics

A variety of evaluation metrics (Hubert and Arabie, 1985; Jaccard, 1901; Rand, 1971; van Rijsbergen, 1979) are available for evaluating SWSI cluster quality. However, most of these will be affected by chance agreement due to the number of clusters used. We therefore use the Adjusted

[1] http://en.wikipedia.org/wiki/Category:Disambiguation_pages

Rand Index (ARI) (Hubert and Arabie, 1985) as our evaluation metric, as it removes the effect of the chance agreement; ARI was used in the SemEval-2013 WSI task. The intuition of ARI is to penalize both false positive and false negative decisions during clustering. The formulation of ARI is as follows:

$$ARI(C_R, C_I) = \frac{RI(C_R, C_I) - E(RI(C_R, C_I))}{\max RI(C_R, C_I) - E(RI(C_R, C_I))}$$

In the formula, C_R is the cluster assignment from the reference sense, and the C_I is the cluster assignment from the SWSI system. $RI(C_R, C_I)$ is the Rand Index between C_R and C_I, and $E(RI(C_R, C_I))$ is the expected value of the RI. A random assignment of word sense (every word is the same sense, or every word is a different sense) will achieve an ARI of 0. The standard ARI ranges from -1 to 1, however we follow the presentation format used in the SemEval 2013 (Navigli and Vannella, 2013) WSI task and multiply the value by 100, to make it range from -100 to +100. In the SemEval 2013 WSI tasks, most teams reported a resulting ARI between 2.5 to 7.1, with a single team has a result of 21.3. However, since the data are different, the ARI numbers can not be directly compared between our tasks and the SemEval 2013 WSI tasks.

In addition to ARI, we also provided Adjusted Mutual Information (AMI) (Vinh et al., 2010) as an extra metric. This is similar to standard Mutual Information based analysis for clustering performance, yet also adjusted to remove the chance agreement. The formulation of AMI is as follows:

$$AMI = \frac{MI - E(MI)}{\max(H(C_R), H(C_I)) - E(MI)}$$

In the formula, $H(C_R)$ is the entropy for cluster assignment from the reference sense, and $H(C_I)$ is the entropy for cluster assignment from the SWSI system. MI is the mutual information between two cluster assignments, and $E(MI)$ is the expectation of the mutual information that can be calculated from the formula described in (Vinh et al., 2010).

Defining the reference cluster for our queries is also a challenge, as asking humans to label the actual word sense would require significant resources. Instead, we use a WordNet-based Word Sense Disambiguation (WSD) approach (Tan, 2014) to label the sense with the human transcript (0% WER) as our reference sense.

For each of the query, we input the entire utterance that the query belongs to into the WSD system to let it label the meaning. The number of reference clusters is decided by the number of senses the WSD system assigned to our input utterances. For example, if the query is "bank", and there are two utterances "I went to deposit money in a nearby bank." and "I was jogging at river bank yesterday.", the WSD system will assign two different meanings to the "bank" word in both sentences. In this way, we label each "bank" word with different meaning, and use that as reference for our experiments.

The reason for inputting the entire utterance is to provide enough context for WSD system to disambiguate the meaning. The number of meanings, which is used in the Analysis section later, is also obtained by the WSD result. If our query word is actually a recognition error (which means it does not occur in the human transcription), the reference sense for that instance is a specific sense of "Wrong Word" that only applies to recognition errors. Note that this is

not part of our training process for distributed representation, but a modification on our ground truth for evaluation. This arrangement can avoid the word that comes from recognition errors being incorrectly assigned to any of the meaning by WSD system, and reduce the quality of our reference. We sampled 100 examples of the reference created by this approach and examined manually to ensure the quality of the reference. We find more than 80% of the labels are correct.

Code

The code that had been used to complete the experiments has been published on on github [2]. The repository contains the data we used to conduct the experiments, and the script we used for all of the experiments reported in this chapter. For each folder, there is an execution script that can runs the entire pipeline. The script relies on the toolkit that will be introduced in the next section. Even without using the script, the data is also on the repository and is available to the research community.

6.4.3 Experimental Setup

Our approach to using distributed representation of utterances for SWSI is straightforward. First, we train the distributed representation using the entire 60 hours of ASR transcription. For each of the utterances that contain the query word, we create a 100-dimension (parameters reported in (Mikolov et al., 2013)) utterance vector, and this vector is projected in the same space that was constructed by the 60 hours of ASR transcription. The utterance vector is trained using standard toolkit[3] available from Google. We then perform repeated bisections clustering (Zhao and Karypis, 2002) on the utterance vector according to a pre-defined number of desired clusters using the CLUTO toolkit (Steinbach et al., 2000), and the MALLET toolkit (McCallum, 2002) for the subsequent LDA-related processing. Since we do not know the ideal number of clusters, we performed a grid-search on possible values; this is reported in section 6.5.1. All parameters are default values unless otherwise specified.

In order to estimate how our SWSI approach compares to the other existing approaches, we also conducted the same experiments using four different baseline systems:

Bag-of-Word (BOW) system: In the BOW system, each utterance is represented by its BOW feature. We then perform repeated bisections clustering on the BOW feature (Pantel and Lin, 2002).

Latent Dirichlet Allocation feature (LDA-feature) system: Instead of using BOW as the feature for each utterance, it first builds an LDA model with 100 topics (in order to match the dimension numbers for word embedding) on the entire 60 hours of testing data. The reason to pick 100 topics is to aim for similar dimension between our LDA features and word embedding features, so the difference is more focused on the modeling algorithm. The repeated bisections clustering uses the topic distribution of utterances as its feature.

Latent Dirichlet Allocation (LDA) system: Described in (Lau et al., 2013), the LDA system trained the topic model only on the utterance in which the query occurs. The number of topics is

[2]https://github.com/jltchiu/SWSI

[3]https://code.google.com/p/word2vec/

the desired cluster number, and each utterance is assigned to the topic that has the highest topical probability. This is different from the LDA-feature system as the LDA-feature system use LDA topic distributions as feature for repeated bisections clustering, while this system use the topic label as clusters.

Hierarchical Dirichlet Processes (HDP) system: Also described in (Lau et al., 2013), the HDP system is trained and clustered in a similar way to the LDA system. However, it does not require any assignment for the topic (cluster) numbers, as the algorithm determines the number of topics automatically. This is similar to the LDA system, which the topic label is the cluster. HDP achieved the best performance in the SemEval-2013 WSI task (Navigli and Vannella, 2013).

We also evaluated our WordNet-based **WSD** system on the ASR transcription. This indicates how an WSD system can perform given a widely-available knowledge source such as WordNet.

We conducted two different sets of experiments. The first set of experiments explores how different approaches perform with different assignments of senses (clusters) on 40% WER data, our expected real-world scenario. The second set of experiments compares how different approaches perform under different WER conditions. This shows how noise introduced by an ASR system affects the SWSI performance for each approach.

6.5 Results

6.5.1 Comparison between WSI approaches

Figure 6.3 shows the ARI performance for our Skip-gram based SWSI system as compared to the four baseline systems on 40% WER data. The WSD system is knowledge-based and indicates the performance achievable with a human-produced resource such as WordNet. None of the other approaches rely on external knowledge. We vary the number of clusters to determine how different approaches interact with the number of clusters. The only exception is the HDP system, as its algorithm will determine the most appropriate number of clusters using a data-driven method. The result shows that, using distributed representation of utterances as feature outperforms other existing feature when processing spoken data on three different level of WERs. This is an encouraging result because it is not only a better modeling technique for clean data, it also shows that it is robust to recognition errors, which is common in ASR systems. If there are further speech understanding related task, distributed representation of utterances can be considered as a robust feature.

The follow sentences are examples of clustering results of word **clear** when number of cluster is set to 3 on 40% WER data:

Cluster one, the meaning of **clear** could be easy to perceive, understand, or interpret:

- The next step is to determine of the nail length and some nails are black some nails are **clear** which makes a little easier to identify the quick or the red heart inside the nail (Wrong meaning)
- And now and do our next one and place it on the opposite side I'm and the reason why we're putting it on the each of the sites first is because then when we put our ribbon on the top it's can a crossover and make a nice **clear** owner and I like when corners or I look all that we've planted out every half

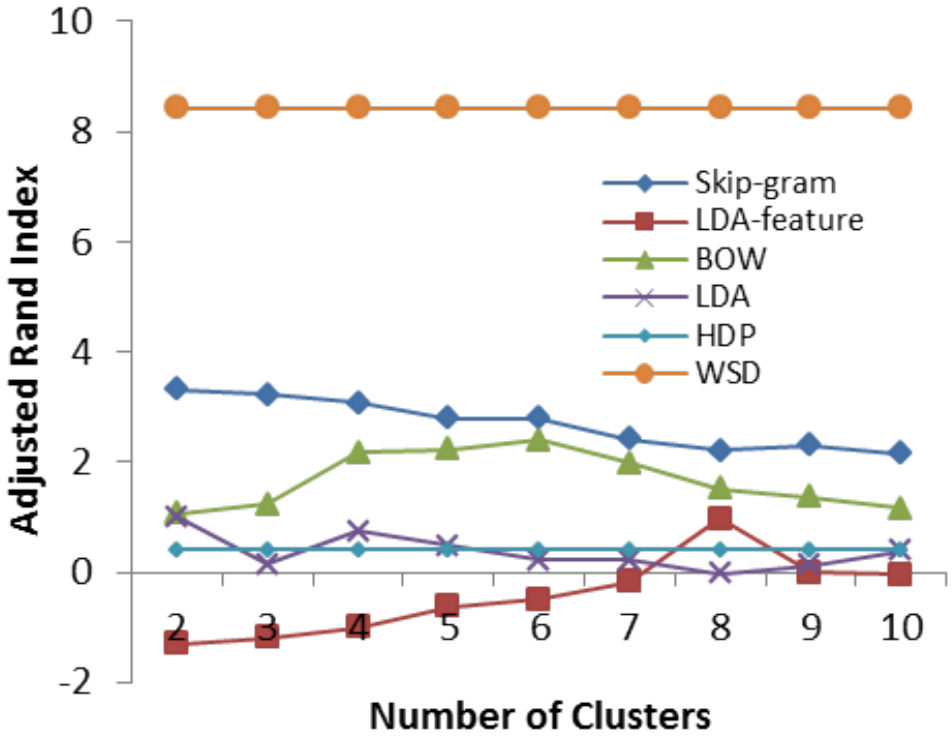

Figure 6.3: ARI Comparison from different approaches with different numbers of clusters on 40% WER data.

- At that point and put it in neutral and let it and give it a quick dad to the throttle just to **clear** the engine now they know holding on it you have to do a release (Wrong meaning)
- When they're demonstrating the right way how to do an exercise they need to make sure that they make a **clear** with our client that they understand what they're doing before the client goes out there and perform hte exercise

Cluster two, the meaning of **clear** could be transparent:

- See inside one of cool my little that off woman dualism use my grid and is move it mainly **clear** should be
- We can use it to separate the top so we have a **clear** defined edge (Wrong meaning)
- So in this case right here with his blueberry peach flying it's **clear** is not hobbling anymore so it's ready to take a and bottle
- This one actually has a **clear** liquin top and bottom

Cluster three, the meaning of **clear** could be free of any obstructions or unwanted objects:

- So in this particular case we're going to be using baking soda salt end a **clear** vinegar (Wrong meaning)
- On ornaments like this that are the characters I usually take a bowl of soapy water and one of **clear**

- You get a really really good sheet to get all the excess glaze off his you only really want a nice thin layer of a **clear** glaze on two piece
- Because that's also a **clear** area that I see quickly

This result shows that, despite not being perfect, we can still see similar meaning of the same word **clear** showing up within the same cluster, even when the WER is as high as 40%.

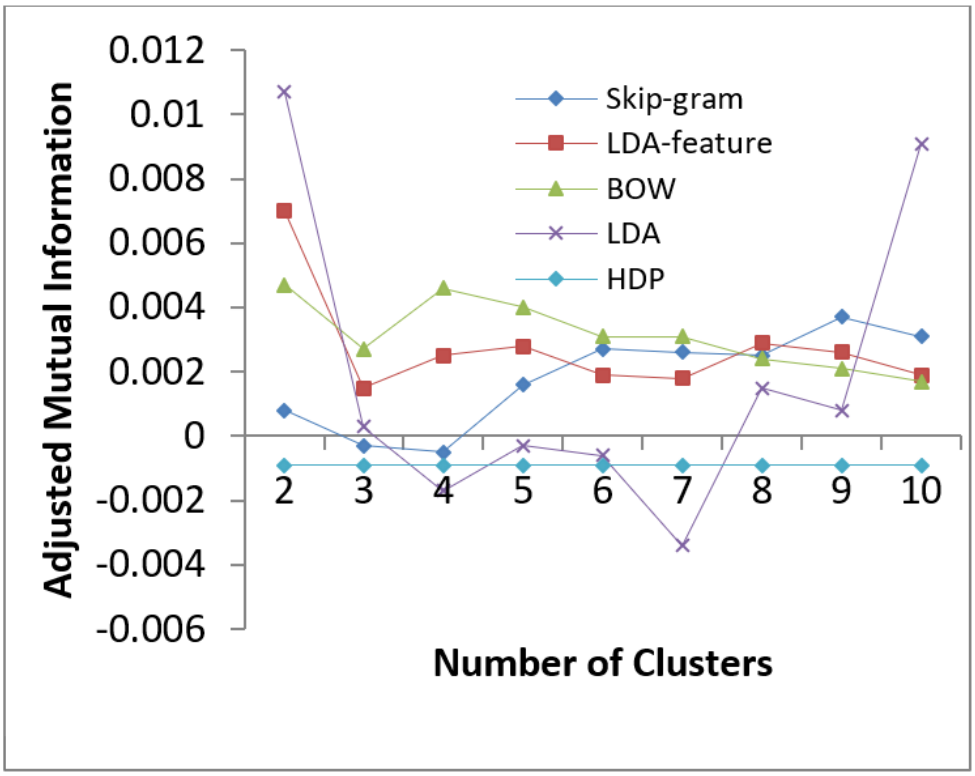

Figure 6.4: AMI Comparison from different approaches with different numbers of clusters on 40% WER data.

The AMI result for 40% WER data is presented in the Figure 6.4. We removed the data line of WSD data because it's value is significantly higher (0.7246) than every other systems, and its not possible to plot that within the same picture. In the AMI metrics, BOW become the system that has the highest overall AMI, while our proposed approach have higher AMI when the number of clusters goes up. However, we believe the difference in these approaches under AMI metric is very limited, as the difference in value is very little.

6.5.2 Comparison between WER

Figure 6.5 shows the comparison between the SWSI systems at different WERs. This result leads us to three conclusions. First, regardless of the varying WER, the Skip-gram based SWSI always achieves the best performance. Second, the LDA-feature system achieves decent performance in the 0% WER condition, but its performance is degraded significantly when noise (i.e. misrecognitions) is present. The noise due to ASR error disrupting the topical distribution, and

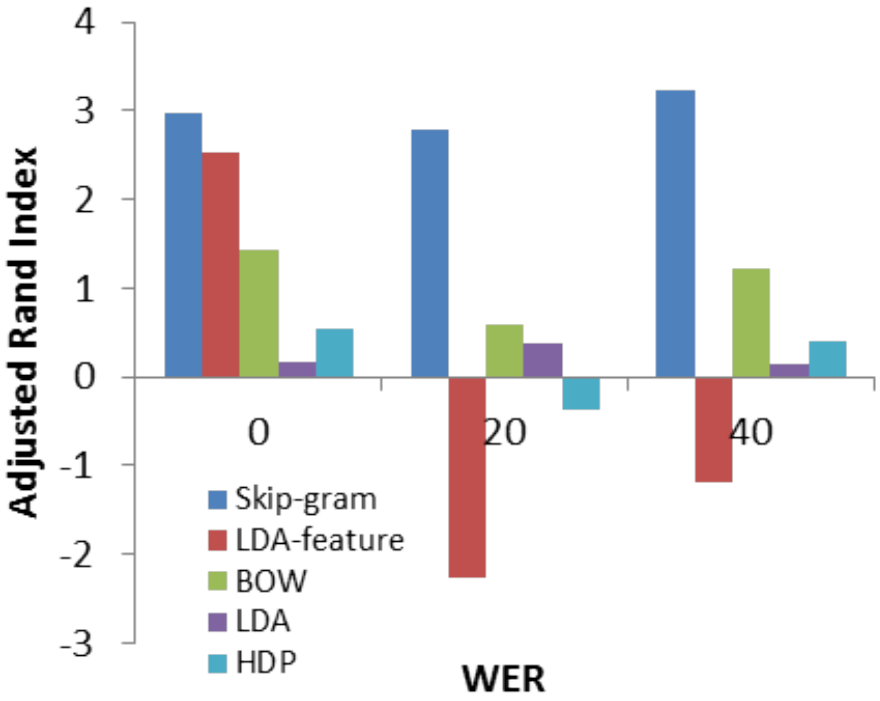

Figure 6.5: ARI Comparison with number of cluster = 3 on different Word Error Rates.

hence degrading the quality of the LDA topical distribution feature. Third, in contrast to general expectation, reducing the WER does not directly translate into a significantly better SWSI performance. We believe this is due to the presence of common locutions. Table 6.2 shows the percentage of the context words around the query that are high frequency (top 1%). Despite the significant difference in WER, the percentage of context consisting of frequently occurring words is similar. This implies that words benefiting from the lower WER may not be the ones that impact the meaning of the content. This also reflects human's conversational behavior, which is weighted towards high-frequency locutions. When people are having conversation, not hearing the locutions clearly has little affect for the understanding of the utterance.

Figure 6.6 shows the comparison in the AMI metrics. Similar to the result we presented in Figure 6.4, the difference between systems are limited. We removed the result of HDP in this figure, as it's performance on 0% WER (0.240) and 20% WER (0.022) are significantly higher than most of the other systems and it will be difficult to plot it in the same picture. One thing worth mentioning is that, even in the AMI setup, the lower WER still does not guarantee better performance, as there are multiple systems with worst performance on 20% WER. Still, the AMI difference between most of the proposed approached are still very limited, except for the approach that can not be plotted in the figure, which has the real difference in the AMI setup.

With all the result presented in 6.5, we found the topic modeling based approach in general have relatively unstable performance, we compared our experiments with classical Topic Modeling experiments reported in (Blei et al., 2003) and found two fundamental differences. First,

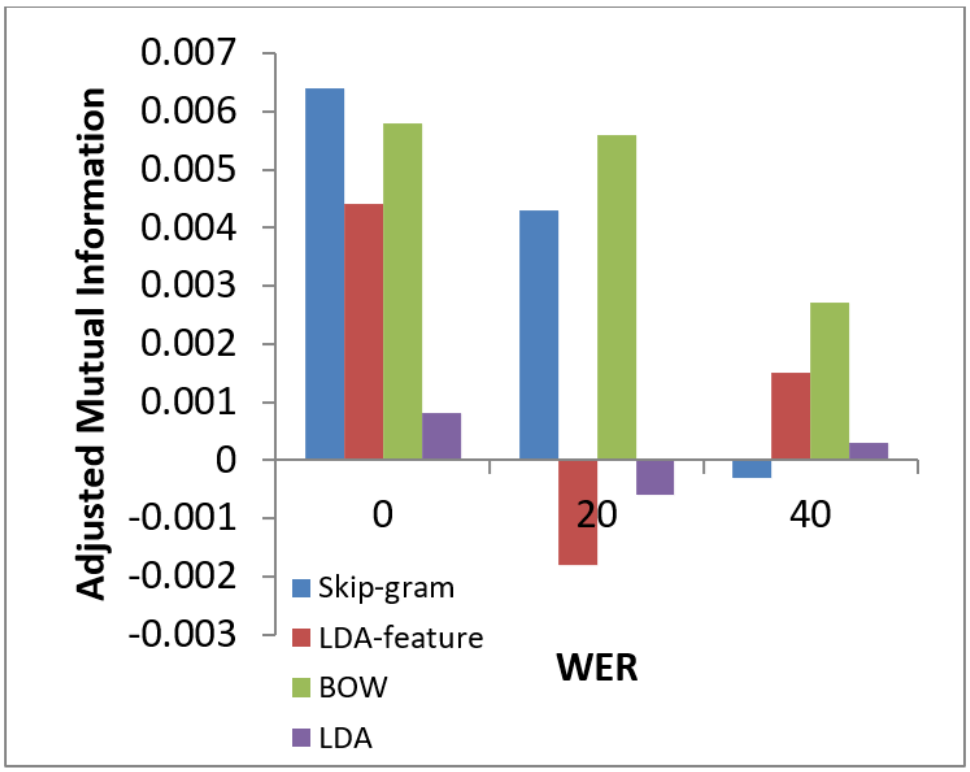

Figure 6.6: AMI Comparison with number of cluster = 3 on different Word Error Rates.

Table 6.2: Percentage of the context that is frequently occurring words.

WER (%)	**40**	**20**	**0**
% of context is frequent word	76.9	78.8	78.1

our data is human speech, which means even if it has 0% WER, the speaker might have stutter or repetition in its utterances, which is still more noisy compare to the news document corpus. Second, we use utterances in a similar way as (Blei et al., 2003) use documents in training LDA models, each utterances is significantly shorter than a news article. As a result, each utterance might not have enough data to train a robust topic model, which causes the unstable performance we observed in this section.

6.6 Analysis

6.6.1 Exploring the Correct Number of Senses

Deciding the correct number of senses/clusters for the query word that shows up in our data is a perennial challenge. In this section, we provide our observations on how the number of reference senses interacts with the cluster numbers in the Skip-gram SWSI system.

Figure 6.5 shows the interaction between the number of assigned clusters and the number

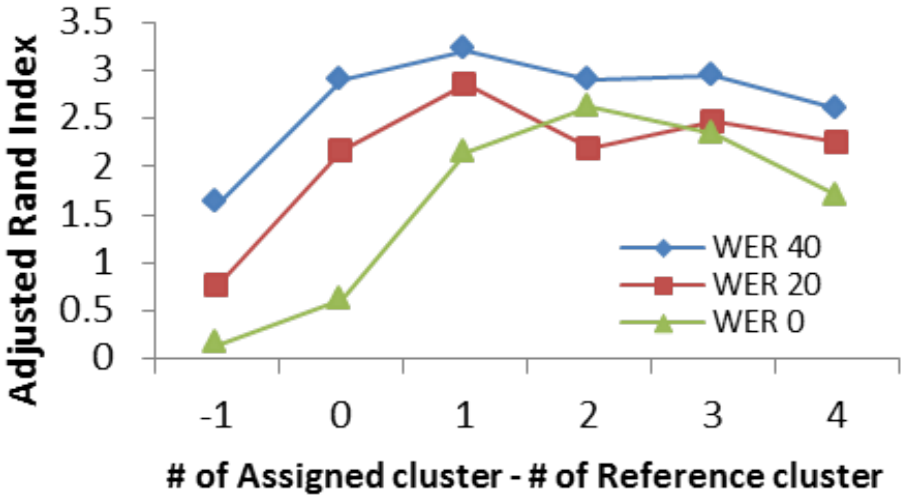

Figure 6.7: ARI Comparison for interaction between the number of assigned and reference clusters.

of reference senses for three different levels of WER. The x axis shows the number of assigned clusters minus the number of reference clusters. The large decrease on the X = 1 is due to multiple instances of queries that have 2 meanings; assigning 1 sense to every word leads to an ARI of 0. According to the result, we observe that assigning 1 or 2 extra clusters compared with the reference sense inventory achieves the best performance.

With further examination, we discovered an interesting phenomenon in the distribution of the results with those extra clusters. When the number of clusters are equal to the number of sense we have, the errors are distributed in every clusters. However, when the extra clusters are available, there will be distinct "correct clusters" that contains the result with same meaning, and the extra cluster becomes "garbage collector" cluster that gathers most the words that meaning can not be figure out clearly with our system. This contribute to better ARI performance as shown in Figure 6.5. Comparing to let the errors being scattered in different clusters, having a few more precise but smaller cluster will improve the ARI performance for our task. This also implies while using our approach, assigning more clusters is generally a good idea comparing to try to set for the exact number of word senses.

6.6.2 Experiments on even higher WER

(Liao et al., 2013) reported ASR experiments on different YouTube data and its performance are around 20% or 40% WER depending on different test data. This work represents the state of the art ASR performance on YouTube data. The reported result are close to the WER we conducted experiments in this chapter, indicating we are showing how SWSI will work with state of the art ASR systems. We believe these are the most meaningful data point, as this reflects how state of the art understanding works with state of the art ASR. Intentionally create extra errors on ASR results or using worse models to reduce the performance of ASR to achieve higher WER results

in artificial errors that might be biased. We believe our experiments in this chapter represents how our approach interact with different quality of ASR system that has no artificial errors included. Intentionally doing experiments on higher WER that is worse than state of the ASR system can be less meaningful, as we won't understand whether the difference in SWSI performance are cause by our artificial errors in ASR systems or our SWSI approach.

6.6.3 Experiments on varying data amounts

Our Skip-gram-based SWSI system achieves good performance on the Word Sense Induction task, but there are some limitations. The distributed representation requires a sufficient amount of training data to produce a stable vector space. We can show this to be the case by varying the amount of data used to train the skip-gram model. We reduced the size of the video dataset to 30 hours and to 10 hours (which contains around 300,000 tokens and 100,000 tokens, respectively) and comparing the performance with the original 60 hours of data. The SWSI performance with Skip-gram system becomes less stable, and in generally inferior to the BOW system when smaller amounts of data are used as can be seen the skip-gram advantage disappears as the amount of training data decreases. We believe this is caused by insufficient training data can not produce a robust distributed representation, hence limiting the performance. The result is shown at Table 6.3, the bold number is the higher ARI between the two.

Table 6.3: Experiments with different amounts of training data

Cluster numbers	**60 Hours**		**30 Hours**		**10 Hours**	
	Skip-gram	**BOW**	**Skip-gram**	**BOW**	**Skip-gram**	**BOW**
2	**3.31**	1.07	1.47	**2.16**	**3.36**	1.91
3	**3.23**	1.23	1.51	**2.72**	1.57	**2.67**
4	**3.07**	2.17	**2.29**	2.25	2.67	**2.71**
5	**2.80**	2.22	1.52	**2.25**	2.28	**2.44**
6	**2.79**	2.39	**1.99**	1.85	**2.07**	1.84

6.7 Discussion

6.7.1 Contribution of this Chapter

The contributions we made in this chapter are:

- We validate Distributional Hypothesis for conversational speech: words that show up in similar contexts tend to be similar to each other, and words that show up in very different context are less likely to be similar to each other, even if they are the same word. This is still valid when there are recognition errors in the context. (6.3.1)
- We describe a technique for separating identical words into multiple groups according to the context. (6.3.2)

- We also show that our approach is more robust to noise compared with existing approaches. This robustness is beneficial for processing real data as most of them will be noisy. (6.5.1)
- We also disprove a common misunderstanding for speech understanding that reducing the WER for speech does not guarantee better understanding, as a 0% WER transcript still has lots of noise (meaningless words) coming from spoken languages. (6.5.2)

The low-level, task-specific contributions we made in this chapter are:

- We present the Spoken Word Sense Induction (SWSI) task, together with a procedure that does not require human labeling for evaluation. (6.2, 6.4)
- We present an unsupervised approach based on Distributed Representation of utterances that can separate the meaning of a specific target word without any label data. (6.3.2)
- We compared our approach with other existing approaches; our approach consistently outperforms existing approaches regardless of different levels of noise in the data. (6.5.1)
- When the correct number of senses is unavailable, assigning a slightly greater number of clusters compared with the correct number of senses for SWSI can still achieve reasonable performance. (6.6.1)
- We identify the limitation of our approaches: when the testing data we used to create the word embedding space are insufficient, we can not have a robust space that represents the relationship between words well. (6.6.3)

6.7.2 Unresolved Issues

Requires data to create word embedding space

In the Analysis section, we describe a limitation of our approach, which is that it requires a minimum amount of data to create the word embedding space. Note that these data are the "testing" data we are going to process, so it still does not require any labeled data to build the space. Still, when the testing data are insufficient, our approach cannot create a robust word embedding space and hence the performance will drop significantly. People have challenged our approach's need to do parameter tuning on development data to achieve good performance. Any algorithm requires information to learn the model, and our approach is not an exception. The testing data we process can help us to create the word embedding space that is customized for the testing domain so that it contains valuable information. Still, we believe that it is far easier to find more unlabeled data to test instead of finding more labeled data for training. We understand that this is a limitation of our approach, yet we believe that it is not a critical one. Our approach is still useful because it provides a new way of feature extraction on data, and we validate this feature is not only useful when the data is clean, but also robust when the data is noisy.

Mapping the cluster to real sense

SWSI only separates instances of target words into multiple clusters according to senses. We do not have labels on each cluster to map our cluster to a specific sense in a dictionary. Also, with our data-driven based approach, usually there will be a cluster that contains all of the instances

for which the system does not know which cluster it should belong to (possibly due to the ambiguous context those words have). As a result, mapping these clusters to the actual meaning of the keyword is not an easy task if no human knowledge is available. We think this is a fundamental problem for all of the WSI-type problem, or even all of the clustering problems. A very straightforward solution for this is to use a small amount of labeled data to bootstrap the system; then all of the clustered data we created can have word sense labels, yet that will no longer be a SWSI task, and there will be an entire research space for those problems. Another possible way of doing this could be using the distribution of words in each cluster to predict what sense it maps to. For example, if a word ended up having two different clusters of meaning, and one cluster is significantly larger than the other one. If we know that the word actually have two senses, and one sense happens more often than the other one, we can map the more common sense to the larger cluster, while the less common sense to the smaller cluster.

The ideal number of clusters

For the WSI community or the clustering research community, there is always a question about how to select the ideal number of clusters. Since we do not have any information available, it is very difficult to identify the correct numbers of clusters for clustering. Some researches (Lau et al., 2013) have proposed identifying the ideal number with a data-driven perspective, like the HDP approach we mentioned as the baseline system in this chapter. However, those approaches do not guarantee that the number of clusters generated will be correct. We think this topic is the core problem in the clustering research, and there could be multiple theses on it; however, it will probably still not be solved. In this thesis, we tend to avoid this question by using the evaluation metrics computed from the distribution of the clusters, so the number of clusters will not have too much impact on our evaluation.

6.7.3 Future Work

Embeddings beyond words

In this chapter, we use the word embedding to represent the relationship between different words and utterances. We believe this embedding space is a good representation to maintain the relationship between different words, and it can be used to capture more valuable information, such as user characteristics. Assuming that the speaker information also comes with the utterances data, we can project every speaker into the vector space as a collection of points in the space, which can be represented as a distribution in the vector space. With the user-specific distribution, and the existing vector spaces that contain the location of words and utterances, we can discover what word or person is possibly more interesting to the user (closer in the space). The fact that we are able to model different units in a same representation makes it easier to capture the relationship between them, especially the unit that is not in the same space like a user and the word he has spoken. With this representation, we believe we can model different forms of communication delivered by humans in a more general and interactive framework, and the information that flows through different forms of communication can be integrated to give us a better understanding of what people want to communicate.

Integration of word embedding and other applications

In this chapter, we explored the use of embedding for the other applications, such as identifying possible ASR recognition errors. The experiments we reported were not successful, yet it still suggests directions for the research forward toward another direction. "Given that we have this embedding that models the relationship between different words, what other task can you apply this information to improve it?" Applying this to other tasks successfully will show that we can leverage the distributed representations for other problems as well.

6.8 Summary

In this chapter, we studied how to leverage the relationship between words and their contexts to create Distributed Representations for utterances for Spoken Word Sense Induction (SWSI). We first present our motivation for studying the SWSI task, since as the STD system will have limitations when the query term is multi-sense word, we need to be able to separate the same word according to its meaning so the user can have better access to the data he or she wants. There are existing methods for performing this task, and we present an approach that can also address this problem but is more robust to noise. Our approach relies on the relationship between the word and its context for creating the word embedding space and separating a single word's multiple word senses. We conduct our experiments on the data collected from YouTube, and there are multiple settings for ASR systems in order to simulate different levels of quality of ASR performance. We compared our approach with approaches that have previously been proposed, and our approach consistently outperforms the others for different values of ASR quality. This shows that our approach is good for tasks particularly like SWSI, where the data are inevitably noisy. The Analysis section discussed the fundamental questions for all of the clustering tasks, the number of clusters, and some of the extra experiments we conducted during the process.

Chapter 7

Conclusion and Future Work

In this thesis, we investigated multiple phenomena that can be observed in human conversations and that to support human understanding of other people's speech. We show that it is possible to extract corresponding feature that support improved automatic processing of speech. First, words that has been spoken recently in conversation is more likely to recur in close proximity. We refer to those as the Word Burst phenomenon. Second, if an identical word had been spoken in the conversation with very similar context, it tends to have similar meaning. In addition, we also identify features presented in automatic processing: when the system store the processed result in different representations, the mismatch between different representations contains information that can provide more accurate at expectation the original data. We investigate the use of these features in three different tasks, Spoken Term Detection (STD), identifying recognition errors and Spoken Word Sense Induction (SWSI). We expect these features will contribute better understanding of human speech, in a variety of applications, such as intent detection(Xu and Sarikaya, 2013) or lie detection(Etcoff et al., 2000).

7.1 Summary of results and contributions

We investigated three different language-independent features in this thesis: Conversational Word Burst, differences in ASR hypotheses, and Distributed Representation of utterances. In the following sections, the results and contributions of our investigations for each feature are summarized.

7.1.1 Conversational Word Burst

For the Word Burst feature, we tried to refine the recognition hypothesis generated from an Automatic Speech Recognition (ASR) system by rescoring hypotheses according to the phenomenon's occurrence in spoken conversation. We designed two different rescoring algorithms, Word Burst rescoring and Unique Penalization rescoring. Word Burst rescoring focuses on words that occur in close proximity, while Unique Penalization focuses on words that appear in a wider context. In order to increase the generality of our approach, we proposed a target expansion technique that can be applied to Word Burst that extends the approach to agglutinative languages, in

which identical word tokens are less likely to reoccur.

We conducted our experiments on multiple languages, and our rescoring algorithm achieved improvements ranging from 2% to 40% relative in ATWV for different languages. For agglutinative languages, by applying the target extension technique, we can improve our performance by 5% relative in ATWV. By carefully examining the results, we found that the gain mostly comes from false alarm reduction, as our algorithm can reduce the score for incorrect decoder output. On average, we can reduce 19% of the false alarms compared with the baseline system. However, we also identify multiple unresolved issues in our approaches. The effect of our rescoring will be reduced if there are low Word Error Rate in pre-rescored hypothesis, as our rescoring will harm the hypotheses that do not follow our assumption. Our approach shows limited performance improvement for words that only occur once in the entire corpus, since our approach relies on the presence of multiple instances. Our approach also requires the recurrence of the same word, which is less likely to happen especially in agglutinative language.

For purpose of generality, we also added experiments on using Word Burst as a feature to identify potential recognition errors in conversational speech. Language-specific information is not used, nor is intensive tuning on development data. Moreover, to our knowledge, this is the first use of information from conversational structure for identifying recognition errors. We also investigated the limitations of the Word Burst feature, and show that under some circumstances, the improvement derived from using the Word Burst feature will be limited. These limitations include: When the systems already have low WER, when the system has limited vocabulary, or when the target language is an agglutinative language.

The contribution we made regarding Word Burst can be separated into two levels. From the high-level perspective, we identify the Word Burst phenomenon as a feature that we can leverage to clean up communicated information. We also validate it as a language-independent phenomenon, as it provides improvements on different languages. From more practical perspective, we show that when performing STD with noisy ASR results, applying our proposed rescoring algorithm achieves performance improvement mostly on false alarm reduction. We can also use Word Burst as a feature to identify recognition error. We think this is one of the applications for which the Word Burst phenomenon can be useful.

7.1.2 Integration of Different Recognition Hypotheses in Spoken Term Detection

We investigate how structural mismatch between different recognition hypotheses can be used to improve the STD performance with system combination. Our assumption is that each recognition hypothesis structure has its own specific error/correct pattern, and we can use system combination on different structure to achieve overall better performance, while the ASR system is trained with the same data. We compared two different popular search methods, Finite-State Transducer (FST) search and Confusion Network (CN) search; each of these correspond to a popular format of recognition hypotheses: lattice and confusion networks. We describe a few techniques to combine the search results.

We performed different sets of experiments to validate our assumptions. All of the experiments were conducted on five different languages to ensure that our approach is general across

languages. The first set of experiments compares the performance of two different search approaches. The FST search has better performance on multi-word query, while the CN search has better performance on single-word query. Then we performed the search combination, and our combination achieved 5% relative improvement compare to the FST search system. Finally, we combined our approach together with the standard multi-ASR combination, and from that the improvement is still additive. In addition to the positive result, we also identified some unresolved issues. The gain from our approach mostly comes from the difference between the two search methods, which are mostly on the multi-word query. The gain on single-word query with our approach is very limited. Another limitation is that, the gain from applying our approach to a single ASR system becomes marginal if we further combine with multiple ASR systems. Different ASR systems have much more significant differences, and those differences could cover the gain we have for a single ASR system.

We identify that there are unique information in each recognition hypotheses. By leveraging those structural mismatch, we can have better understanding of the original information that are processed by ASR system. It is a phenomenon that is not intrinsic to data, as we can gain improvement on different language and different decoder configuration. The improvement we get is mostly related to the queries which both search algorithms differ, which are multi-word queries. The presented experiments represent a way of leveraging the unique information contained in each recognition hypotheses structure.

7.1.3 Distributed Representation of utterance

Finally we discussed is the extraction of better features for the SWSI task. By creating a word embedding space that follows the Distributional Hypothesis, we can project words in our target audio into a high-dimensional vector space. We then create an utterance vector that represents every utterance in the same space, and use the utterance vector to cluster multiple instances of the same word. For every instance of a target word for which we want to perform SWSI, we represent each instance of the target word as the utterance vector it belongs to, and the clustering is based on the utterance vector. This approach enables us to separate different meanings of the same word, originally a single point in the vector space.

The experiments were conducted on 60 hours of YouTube video. We selected YouTube video as our data because it constitutes real-world data, instead of carefully curated, clean data. The YouTube video also comes with user-uploaded subtitles that can be used as the reference transcription to evaluate the quality of our ASR system. We compare our approach with multiple existing baselines on three different levels of ASR performance, 40% WER, 20% WER, and 0% WER.

Our approach outperforms every baseline systems regardless of ASR performance. The positive result indicate it is a feature that can provide good performance. Moreover, it is very robust to noise, since its good performance is still present when the data become noisy. This robustness makes it a good feature to use on spoken data. We also found that, by deciding the number of clusters to be slightly more than the number actual senses, we can achieve better performance comparing to assigning the cluster number equal to the number of sense. A few limitations of our approach are also discussed. When the data are insufficient, the created word embedding will not have good performance. Also, with our approach, there is still not a good way to automatically

map our clusters to the actual meaning of words.

We separate the contribution we made into two parts. With respect to the high-level contribution, we showed that the Distributional Hypothesis can be used to create a representation that models the relationship between different words. Moreover, to extend our representation to a larger unit (in this chapter, it is from word to utterance), we are able to separate different instances of the same word. Our work shows that our approach can help to robustly identify the meaning of the word (SWSI). When we use our feature on the actual task, there are more empirical contributions. First, we present the SWSI task and introduce a way to evaluate it without human labeling for evaluation. Second, we present an unsupervised approach that can be used to separate the meanings of a specific word. Our approach gives better performance and is more robust compared with the previously reported approach. Our result suggests that for future speech processing research, the feature we propose can separate different meanings of a word in a robust way. Using context is the primary way of separating meaning, yet building "word embedding" space with context could be a better way of using context compare with LDA or bag of words as the feature.

7.2 Future Work

In this thesis, we described different features that can be used to search or understand spoken data. The feature we propose are not depending on the characteristics of specific languages, rather they are features that are present in the natural structure of conversation. We believe this leads in an interesting direction, which is leveraging features that are more generic and independent from simply the content of conversation. In addition to what we described in this thesis, we provide some additional directions that we believe could constitute the next step towards fuller incorporation of conversational features into the tasks we studied.

Identify the semantic unit

Our work on Word Burst uses identical word token as the key for the algorithm. The reason why we use identical word is because it is most likely to have the same semantics. However, we believe that, even if the word tokens are different, if the semantic units are the same, we should still observe the recurrence of the semantic unit. If there is a way that can convert a sentence into a sequence of semantic units, lots of the language processing algorithms such as Word Burst rescoring can be expand to the semantic unit. Also, if the semantic unit is defined, then language should not be an issue, since the words in different languages should be able to convert to this fix set of semantic units. One possible way of approaching this problem is to first define a fixed set of semantic units, then try to describe everything human can say within that limited set of units. We believe this is a challenging future directions, yet finding out a good way to represent semantics will be a revolutionary progress in the research community. One possibility would be an adaptation of the word embedding approach to this problem.

Beyond the identical word

In Chapter 4, we tried to leverage the recurrence of identical words for recognition hypothesis rescoring. A very reasonable extension is to extend this approach to model the relationship between the word. Since there are properties that can describe the occurrence of the identical word, the co-occurrence of different words will likely contain useful information. We made several attempts in Chapter 4 to identify this property, but we have not yet achieved a successful result. In our opinion, the concurrence of different words can be addressed in multiple ways. We can try to model the occurrence in the close distance like how the n-gram language mode works, or consider a more global occurrence like how topic models affect the understanding of the data. In either way, if we can successfully model the relationship between different words, we might even be able to process words that are rare or unknown, as the context will still provide us much information. There are existing tools like word2vec (Mikolov et al., 2013) that are trying to model this relationship, yet the result of their modeling is hard to interpret and use in other tasks, as we show in our experiments.

Integration beyond Spoken Language

Although this thesis has mostly focused on spoken language, leveraging the phenomena we discussed in this thesis to different forms of communication such as visual or tactile communication can help us to gain a better understanding of the information that is intended to be delivered. One benefit for integrating from multiple sources of information is that it can also be used to detect the mismatch between different forms of communication, which can possibly identify lying, a phenomenon we work very hard to detect within a single form of communication. When someone is talking to you yet their facial expression seems nervous, the understanding of their spoken language should be changed. If the conversational system is part of a robot, when receiving a petting pat on the head, the system should also know that it is probably talking to someone who is friendly. Communication itself is a multi-dimensional interaction with people, and only by integrating the information from each perspective can we reconstruct the whole picture for the information that is intended to be delivered. This will also bridge the research from different fields together to attempt to understand how communication really works between humans.

Smart integration strategy System combination is always a way to provide a better result for an existing task. Still, if an automatic or a more optimal way of fusing the systems can be explored, then we can save significant engineering efforts. To date, there have not been many research efforts addressing this question, yet we believe this will yield significant impact, especially in industrial applications. After all, combination of systems is one of the simplest ways of leveraging the strength of multiple systems, and a smart strategy for doing so can be helpful on many problems that require system combination to push the performance. A possible way for addressing this problem could be designing some objective function, and automatically tune the system like doing gradient descend on parameters to optimize for the objective function.

Embeddings beyond words

We present our approach in Chapter 6 by projecting words and utterances into a word embedding space. The embedding can preserve the relationship between different units that have been projected into the space, yet in our work we only focus on words and utterances. We believe that different kinds of units can be introduced and also projected into the space, to obtain the relationship between different units. For example, if the speaker information from all of the utterances are all available, then we can project the speaker into the space, and all of the utterances/words that are more related to that point can be considered as a possible interest for that speaker. In this way, we can try to project information from different dimensions into the same space. Hence, our embedding can cover more complicated information compared with the original word/utterance-only information. The ability to make the information from different dimensions interact within the same embedding space is the strength of this approach, yet we believe that it can be a good way to incorporate multiple forms of information for better understanding of communication. Another approach could be integrating time or location information into the embeddings. If the words that are used to train a model has time or location labeled, then even the same word spoken at different location or time will be different points in the space. For example, the word "football" could be used to create embeddings, yet if we have location information available, we might know that the "football" spoken in Europe might have a very different meaning from the "football" spoken in USA. That will enable us model context that are beyond the content of conversation.

Integration of word embedding and other applications

The only application to which we had applied word embedding successfully in this thesis is the SWSI task. We did attempt to use embedding for other applications (which is reported in Chapter 6), yet we could not achieve any significant improvements. Research questions remain about what we can do with this embedding other than the SWSI task. In the Related Work section, we discuss multiple other tasks that had been presented by researchers to apply word embedding, such as statistical language modeling, parsing, tagging or machine translation. The word embedding is considered as a good model for capture the relationship between different words. The model gives a the word a continuous space to represent its relationship, instead of the dictionary type information which is a hard decision (like the words are synonyms or antonyms). Still, with the noise coming from the spoken data, applying the embedding trained from noisier data will be a challenge we need to address in the future. After all, despite the fact that there will be more data available, most of the data we collect will not be clean. Our experiments show that word embedding is ideal for processing noisy data in SWSI compare to other modeling approaches, so applying this embedding to other applications should be a reasonable next step to pursue.

Identifying Intention

Our thesis has addressed different phenomena that can help us to extract more useful information, yet there is more to human communication. Why do humans seek to communicate? There is probably always an "intention" behind any activity, and that can be a next step after we achieve

understanding about these phenomena. When the human has different intentions in mind, the actions they demonstrate will probably be different, so is it possible to model that? Or can we predict the intention based on the phenomenon that we can observe from their action/speech? We believe that there is an entire space that has not yet been explored, and many research opportunities lie within it. The phenomena we presented in this thesis can be considered as features of human activities, and we believe that it is useful to better understand human activities according to these features. In order to achieve this goal, we can label utterances with pre-defined intent labels, and then train statistical classifier with different types of features. Word based feature such as bag of word could be used, and additional features including information about conversation, speaker, environment, context, topics can also be used for training the classifier. In this setup, we will be able to identify other people's intention as long as their intent is one of the label that our classifier can predict. In the end, its really a way of making our systems more intelligent by using better feature from better sources.

Bibliography

Agirre, E., Martínez, D., De Lacalle, O. L., and Soroa, A. (2006). Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 89–96. Association for Computational Linguistics.

Airoldi, E. M., Fienberg, S. E., and Xing, E. P. (2007). Mixed membership analysis of genome-wide expression data. *arXiv preprint arXiv:0711.2520*.

Allauzen, A. (2007). Error detection in confusion network. In *INTERSPEECH*, pages 1749–1752.

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.

Anguera, X., Skácel, M., Vorwerk, V., and Luque, J. (2013). The telefonica research spoken web search system for mediaeval 2013. In *MediaEval*.

Apidianaki, M. (2008). Translation-oriented word sense induction based on parallel corpora. In *Language Resources and Evaluation (LREC)*.

Banerjee, S. and Rudnicky, A. I. (2004). Using simple speech–based features to detect the state of a meeting and the roles of the meeting participants.

Barnett, J. (1973). A vocal data management system. *Audio and Electroacoustics, IEEE Transactions on*, 21(3):185–188.

Baum, L. E., Eagon, J., et al. (1967). 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.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, pages 1554–1563.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, pages 164–171.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Benıtez, M., Rubio, A., Garcıa, P., and de la Torre, A. (2000). Different confidence measures for

word verification in speech recognition. *Speech Communication*, 32(1):79–94.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Brody, S. and Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111. Association for Computational Linguistics.

Brown, E. W., Srinivasan, S., Coden, A., Ponceleon, D., Cooper, J. W., and Amir, A. (2001). Toward speech as a knowledge resource. *IBM Systems Journal*, 40(4):985–1001.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

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

Chelba, C., Hazen, T. J., and Saraçlar, M. (2008). Retrieval and browsing of spoken content. *Signal Processing Magazine, IEEE*, 25(3):39–49.

Chen, G., Khudanpur, S., Povey, D., Trmal, J., Yarowsky, D., and Yilmaz, O. (2013a). 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.

Chen, G., Yilmaz, O., Trmal, J., Povey, D., and Khudanpur, S. (2013b). 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.

Chen, Y.-N., Chen, C.-P., Lee, H.-Y., Chan, C.-A., and Lee, L.-S. (2011). 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.

Chiu, J., Miao, Y., Black, A. W., and Rudnicky, A. I. (2015). Distributed representation-based spoken word sense induction. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Chiu, J. and Rudnicky, A. (2014). Lacs system analysis on retrieval models for the mediaeval 2014 search and hyperlinking task.

Chiu, J. and Rudnicky, A. I. (2013). Using conversational word bursts in spoken term detection. In *INTERSPEECH*, pages 2247–2251.

Chiu, J., Wang, Y., Trmal, J., Povey, D., Chen, G., and Rudnicky, A. (2014). Combination of fst and cn search in spoken term detection. In *Proc. Interspeech*.

Church, K. W. and Gale, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, 1(02):163–190.

Clark, P., Fellbaum, C., Hobbs, J. R., Harrison, P., Murray, W. R., and Thompson, J. (2008). Augmenting wordnet for deep understanding of text. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 45–57. Association for Computational Linguistics.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep

neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42.

Dai, A. M., Olah, C., and Le, Q. V. (2015). Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.

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

Dempster, A. P., Laird, N. M., Rubin, D. B., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38.

Di Marco, A. and Navigli, R. (2013). Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754.

Dorow, B., Widdows, D., Ling, K., Eckmann, J.-P., Sergi, D., and Moses, E. (2004). Using curvature and markov clustering in graphs for lexical acquisition and word sense discrimination. *arXiv preprint cond-mat/0403693*.

Doyle, G. and Elkan, C. (2009). Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 281–288. ACM.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Etcoff, N. L., Ekman, P., Magee, J. J., and Frank, M. G. (2000). Lie detection and language comprehension. *Nature*, 405(6783):139.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Finke, M., Geutner, P., Hild, H., Kemp, T., Ries, K., and Westphal, M. (1997). 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.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Firth, J. R. (1968). *Selected papers of JR Firth, 1952-59*. Indiana University Press.

Fiscus, J. G., Ajot, J., Garofolo, J. S., and Doddington, G. (2007). Results of the 2006 spoken term detection evaluation. In *Proc. SIGIR*, volume 7, pages 51–57. Citeseer.

Fu, Q.-J., Zeng, F.-G., Shannon, R. V., and Soli, S. D. (1998). Importance of tonal envelope cues in chinese speech recognition. *The Journal of the Acoustical Society of America*, 104(1):505–510.

Harris, Z. S. (1954). Distributional structure. *Word*.

Hazen, T. J., Shen, W., and White, C. (2009). Query-by-example spoken term detection using phonetic posteriorgram templates. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 421–426. IEEE.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Hinton, G. E. (1984). Distributed representations.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Jaccard, P. (1901). *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz.

James, D. A. and Young, S. J. (1994). A fast lattice-based approach to vocabulary independent wordspotting. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 1, pages I–377. IEEE.

Jansen, A., Church, K., and Hermansky, H. (2010). Towards spoken term discovery at scale with zero resources. In *INTERSPEECH*, pages 1676–1679.

Jansen, A., Dupoux, E., Goldwater, S., Johnson, M., Khudanpur, S., Church, K., Feldman, N., Hermansky, H., Metze, F., Rose, R., et al. (2013). A summary of the 2012 jhu clsp workshop on zero resource speech technologies and models of early language acquisition.

Jelinek, F. (1997). *Statistical methods for speech recognition*. MIT press.

Jelinek, F., Merialdo, B., Roukos, S., and Strauss, M. (1991). A Dynamic Language Model for Speech Recognition. In *HLT*, volume 91, pages 293–295.

Jiang, H. (2005). Confidence measures for speech recognition: A survey. *Speech communication*, 45(4):455–470.

Karakos, D., Schwartz, R., Tsakalidis, S., Zhang, L., Ranjan, S., Ng, T., Hsiao, R., Saikumar, G., Bulyko, I., Nguyen, L., et al. (2013). Score normalization and system combination for improved keyword spotting. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 210–215. IEEE.

Khapra, M. M., Shah, S., Kedia, P., and Bhattacharyya, P. (2009). Projecting parameters for multilingual word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 459–467. Association for Computational Linguistics.

Knill, K., Gales, M., Rath, S., Woodland, P., Zhang, C., and Zhang, S.-X. (2013). 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.

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

Kupiec, J. (1989). 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.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Lau, J. H., Cook, P., and Baldwin, T. (2013). unimelb: Topic modelling-based word sense induction. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 307–311.

Lee, H.-y., Zhang, Y., Chuangsuwanich, E., and Glass, J. (2014). Graph-based re-ranking using acoustic feature similarity between search results for spoken term detection on low-resource languages. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Liao, H., McDermott, E., and Senior, A. (2013). 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.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.

Lin, D. and Pantel, P. (2001). Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Liu, Y.-C., Liu, M., and Wang, X.-L. (2012). *Application of Self-Organizing Maps in Text Clustering: A Review*. INTECH Open Access Publisher.

Mamou, J., Cui, J., Cui, X., Gales, M. J., Kingsbury, B., Knill, K., Mangu, L., Nolden, D., Picheny, M., Ramabhadran, B., et al. (2013). 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.

Mamou, J., Ramabhadran, B., and Siohan, O. (2007). 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.

Mangu, L., Brill, E., and Stolcke, A. (1999). Finding consensus among words: lattice-based word error minimization. In *Eurospeech*. Citeseer.

Mangu, L., Brill, E., and Stolcke, A. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

Mangu, L. and Padmanabhan, M. (2001). 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.

Mangu, L., Soltau, H., Kuo, H.-K., Kingsbury, B., and Saon, G. (2013). Exploiting diversity for spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8282–8286. IEEE.

McCallum, A. K. (2002). {MALLET: A Machine Learning for Language Toolkit}.

Miao, Y. and Metze, F. (2013). Improving low-resource CD-DNNHMM using dropout and multilingual DNN training. In *Proc. Interspeech*, pages 2237–2241.

Miao, Y., Metze, F., and Rawat, S. (2013). Deep maxout networks for low-resource speech recognition. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 398–403. IEEE.

Mikolov, T. (2012). Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Miller, D. R., Kleber, M., Kao, C.-L., Kimball, O., Colthurst, T., Lowe, S. A., Schwartz, R. M., and Gish, H. (2007). Rapid and accurate spoken term detection. In *INTERSPEECH*, pages 314–317.

Narasimhan, K., Karakos, D., Schwartz, R., Tsakalidis, S., and Barzilay, R. (2014). Morphological segmentation for keyword spotting.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Navigli, R. and Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 116–126. Association for Computational Linguistics.

Navigli, R. and Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):678–692.

Navigli, R. and Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

Navigli, R. and Vannella, D. (2013). Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 193–201.

Ortmanns, S., Ney, H., and Aubert, X. (1997). A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech & Language*, 11(1):43–72.

Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

Pedersen, T. (2013). Duluth: Word sense induction applied to web page clustering. *Atlanta, Georgia, USA*, page 202.

Pinto, D., Rosso, P., and Jimenez-Salazar, H. (2007). Upv-si: Word sense induction using self term expansion. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 430–433. Association for Computational Linguistics.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The Kaldi speech recognition toolkit. In *Proc. ASRU*, pages 1–4.

Povey, D., Hannemann, M., Boulianne, G., Burget, L., Ghoshal, A., Janda, M., Karafiát, M., Kombrink, S., Motlicek, P., Qian, Y., et al. (2012). Generating exact lattices in the WFST framework. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4213–4216. IEEE.

Qin, L. (2013). *Learning Out-of-Vocabulary Words in Automatic Speech Recognition*. PhD thesis, Citeseer.

Rabiner, L. R. and Juang, B.-H. (1993). *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Rocchio, J. J. (1971). Relevance feedback in information retrieval.

Rosenfeld, R. and Huang, X. (1992). Improvements in stochastic language modeling. In *Proceedings of the workshop on Speech and Natural Language*, pages 107–111. Association for Computational Linguistics.

San-Segundo, R., Pellom, B., Hacioglu, K., Ward, W., and Pardo, J. M. (2001). Confidence measures for spoken dialogue systems. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 393–396. IEEE.

Schütze, H. (1998). Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

Steinbach, M., Karypis, G., Kumar, V., et al. (2000). A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, MA.

Stoyanchev, S., Salletmayr, P., Yang, J., and Hirschberg, J. (2012). Localized detection of speech recognition errors. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 25–30. IEEE.

Tan, L. (2014). Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]. https://github.com/alvations/pywsd.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476).

Tejedor, J., Toledano, D. T., Lopez-Otero, P., Docio-Fernandez, L., Garcia-Mateo, C., Cardenal, A., Echeverry-Correa, J. D., Coucheiro-Limeres, A., Olcoz, J., and Miguel, A. (2015). Spoken term detection albayzin 2014 evaluation: overview, systems, results, and discussion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):1–27.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Tulving, E. and Schacter, D. L. (1990). Priming and human memory systems. *Science*, 247(4940):301–306.

Tur, G., Deoras, A., and Hakkani-Tur, D. (2013). Semantic Parsing Using Word Confusion Networks With Conditional Random Fields. In *Proc. of the INTERSPEECH*.

Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

van Rijsbergen, C. (1979). *Information Retrieval. 1979*. Butterworth.

Vergyri, D., Shafran, I., Stolcke, A., Gadde, V. R. R., Akbacak, M., Roark, B., and Wang, W. (2007). The SRI/OGI 2006 spoken term detection system. In *INTERSPEECH*, pages 2393–2396. Citeseer.

Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854.

Wang, H., Lee, T., Leung, C.-C., Ma, B., and Li, H. (2013). Using parallel tokenizers with dtw matrix combination for low-resource spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8545–8549. IEEE.

Wegmann, S., Faria, A., Janin, A., Riedhammer, K., and Morgan, N. (2013). 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.

Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Xu, H., Povey, D., Mangu, L., and Zhu, J. (2011). Minimum Bayes Risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language*, 25(4):802–828.

Xu, P. and Sarikaya, R. (2013). Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 78–83. IEEE.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Young, S., Hauptmann, A. G., Ward, W. H., Smith, E. T., and Werner, P. (1989). High level knowledge sources in usable speech recognition systems. *Communications of the ACM*, 32(2):183–194.

Yu, S.-I., Jiang, L., and Hauptmann, A. (2014). Instructional videos for unsupervised harvesting and learning of action examples. In *Proceedings of the ACM International Conference on Multimedia*, pages 825–828. ACM.

Zhang, X., Trmal, J., Povey, D., and Khudanpur, S. (2014). Improving deep neural network acoustic models using generalized maxout networks. *submitted to ICASSP*.

Zhang, Y. and Glass, J. R. (2009). Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 398–403. IEEE.

Zhao, Y. and Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM.

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.

Zou, W. Y., Socher, R., Cer, D. M., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.