Modelling Out-of-Vocabulary Words for Robust Speech Recognition

by

Issam Bazzi

S.M., Massachusetts Institute of Technology (1997)
B.E., American University of Beirut, Beirut, Lebanon (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author .....
Department of Electrical Engineering and Computer Science
May, 2002

Certified by .....
James Glass
Principal Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Committee on Graduate Students

Modelling Out-of-Vocabulary Words for Robust Speech Recognition

by

Issam Bazzi

Submitted to the Department of Electrical Engineering and Computer Science
on May, 2002, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Abstract

This thesis concerns the problem of unknown or *out-of-vocabulary* (OOV) words in continuous speech recognition. Most of today's state-of-the-art speech recognition systems can recognize only words that belong to some predefined *finite* word vocabulary. When encountering an OOV word, a speech recognizer erroneously substitutes the OOV word with a similarly sounding word from its vocabulary. Furthermore, a recognition error due to an OOV word tends to spread errors into neighboring words; dramatically degrading overall recognition performance.

In this thesis we propose a novel approach for handling OOV words within a single-stage recognition framework. To achieve this goal, an explicit and detailed model of OOV words is constructed and then used to augment the closed-vocabulary search space of a standard speech recognizer. This OOV model achieves open-vocabulary recognition through the use of more flexible subword units that can be concatenated during recognition to form new phone sequences corresponding to potential new words. Examples of such subword units are phones, syllables, or some automatically-learned multi-phone sequences. Subword units have the attractive property of being a closed set, and thus are able to cover any new words, and can conceivably cover most utterances with partially spoken words as well.

The main challenge with such an approach is ensuring that the OOV model does not absorb portions of the speech signal corresponding to in-vocabulary (IV) words. In dealing with this challenge, we explore several research issues related to designing the subword lexicon, language model, and topology of the OOV model. We present a dictionary-based approach for estimating subword language models. Such language models are utilized within the subword search space to help recognize the underlying phonetic transcription of OOV words. We also propose a data-driven iterative bottom-up procedure for automatically creating a multi-phone subword inventory. Starting with individual phones, this procedure uses the maximum mutual information principle to successively merge phones to obtain longer subword units.

The thesis also extends this OOV approach to modelling multiple classes of OOV words. Instead of augmenting the word search space with a single model, we add several models, one for each class of words. We present two approaches for designing the OOV word classes. The first approach relies on using common part-of-speech tags. The second approach is a data-driven two-step clustering procedure, where the first step uses agglomerative clustering to derive an initial class assignment, while the second step uses iterative clustering to move words from one class to another in order to reduce the model perplexity.

We present experiments on two recognition tasks: the medium-vocabulary spontaneous speech JUPITER weather information domain and the large-vocabulary broadcast news HUB4 domain. On the JUPITER task, the proposed approach can detect 70% of the OOV words with a false alarm rate of less than 3%. At this operating point, the word error rate (WER) on the IV utterances degrades slightly (from 10.9% to 11.2%) while the overall WER decreases from 17.1% to 16.4%. Furthermore, the OOV model achieves a phonetic error rate of 31.2% on correctly detected OOV words. Similar performance is achieved on the HUB4 domain, both in detecting OOV words as well as in reducing the overall WER.

In addition, the thesis examines an approach for combining OOV modelling with recognition confidence scoring. Since these two methods are inherently different, an approach that combines the techniques can provide significant advantages over either of the individual methods. In experiments in the JUPITER weather domain, we compare and contrast the two approaches and demonstrate the advantage of the combined approach. In comparison to either of the two individual approaches, the combined approach achieves over 25% fewer false acceptances of incorrectly recognized keywords (from 55% to 40%) at a 98% acceptance rate of correctly recognized keywords.

Thesis Supervisor: James Glass
Title: Principal Research Scientist

Acknowledgments

My deepest gratitude goes to my thesis advisor, Jim Glass, for his help and support throughout the four years I spent at SLS. Jim always guided me at every stage of my graduate studies, and gave me great ideas and suggestions. Without him, this thesis would not have been possible.

I would like to thank my thesis committee, Victor Zue and Trevor Darrell. Victor Zue had very stimulating questions and provided insightful feedback to various aspects of my thesis work. Trevor Darrell provided many valuable suggestions and brought a different perspective to my research work.

I am grateful to Jim, Victor, and all members of SLS who provided the great research environment that helped me accomplish my goals. SLS is a place I am going to miss soon after leaving.

I would like to thank Lee Hetherington for his help on finite-state transducers. He answered all my question and helped establishing the tools used in my research. I would also like to thank TJ Hazen who helped me in understanding many aspects of the SUMMIT system. It was a great pleasure to work with TJ on the topic of integrating OOV modelling and confidence scoring.

A very special thank goes to my officemate Karen Livescu. Karen was very helpful in many ways, from the long discussions we had on FSTs and various research topics, to her careful reviews of many of my papers as well as parts of this thesis.

I would also like to thank my officemates Min Tang and Sterling Crockett. Thanks to Stephanie Seneff, Chao Wang, Han Shu, Michelle Spina, Joe Polifroni, and Scott Cyphers, as well as to Sally Lee and Vicky Palay.

Finally, I would like to thank my family and friends who are always there for me. It has been a long and challenging experience, and I know that I would have never been able to go through it without their support and guidance.

This dissertation is based upon work supported by DARPA under contract N66001-99-1-8904, monitored through Naval Command, Control and Ocean Surveillance Center, by the National Science Foundation under Grant No. IRI-9618731, and by a graduate fellowship from Microsoft Corporation.

Contents

1 Introduction **17**
- 1.1 Introduction . . . 17
- 1.2 Thesis Goals . . . 20
- 1.3 Outline . . . 22

2 Experimental Setup **25**
- 2.1 Introduction . . . 25
- 2.2 The SUMMIT Recognition System . . . 25
  - 2.2.1 Segmentation . . . 27
  - 2.2.2 Acoustic Modelling . . . 27
  - 2.2.3 Lexical Modelling . . . 27
  - 2.2.4 Language Modelling . . . 28
  - 2.2.5 Recognition . . . 28
- 2.3 Finite-State Transducers . . . 29
- 2.4 The Corpora . . . 29
  - 2.4.1 The JUPITER Corpus . . . 30
  - 2.4.2 The HUB4 Corpus . . . 31
- 2.5 Summary . . . 31

3 Survey and Analysis **33**
- 3.1 Introduction . . . 33
- 3.2 Approaches . . . 33
  - 3.2.1 Vocabulary Optimization . . . 33
  - 3.2.2 Confidence Scoring . . . 34
  - 3.2.3 Multi-Stage Subword Recognition . . . 34
  - 3.2.4 Filler Models . . . 35
- 3.3 Prior Research . . . 35
- 3.4 Vocabulary Growth and Coverage . . . 39
- 3.5 Analysis of OOV Words . . . 41
  - 3.5.1 Length of OOV words . . . 42
  - 3.5.2 Types of OOV Words . . . 44
- 3.6 Summary . . . 45

4 Modelling OOV Words for Robust Speech Recognition **47**
- 4.1 Introduction . . . 47
- 4.2 The General Framework . . . 48
- 4.3 Modelling Out-Of-Vocabulary Words . . . 49

4.3.1 The IV Search Network . . . 50
4.3.2 The OOV Search Network . . . 51
4.3.3 The Hybrid Search Network . . . 52
4.3.4 The Probability Model . . . 53
4.3.5 Comparison with Other Approaches . . . 55
4.4 OOV Model Configurations . . . 56
4.4.1 The Corpus Model . . . 56
4.4.2 The Dictionary Model . . . 56
4.4.3 The Oracle OOV Model . . . 57
4.5 Applying Topology Constraints . . . 58
4.5.1 OOV Length Constraints . . . 58
4.5.2 The Complement OOV Model . . . 59
4.6 Performance Measures . . . 61
4.6.1 Detection and False Alarm Rates . . . 61
4.6.2 Recognition Accuracy . . . 62
4.6.3 Location Accuracy . . . 62
4.6.4 Phonetic Accuracy . . . 63
4.7 Experimental Setup . . . 63
4.7.1 JUPITER Baseline . . . 63
4.7.2 HUB4 Baseline . . . 65
4.8 JUPITER Results . . . 66
4.8.1 Detection Results . . . 66
4.8.2 Impact on Recognition Performance . . . 68
4.8.3 Locating OOV Words . . . 70
4.8.4 Phonetic Accuracy . . . 72
4.8.5 Imposing Topology Constraints . . . 75
4.9 HUB4 Results . . . 76
4.10 Conclusions . . . 77
4.11 Summary . . . 79

5 Learning Units for Domain-Independent OOV Modelling 81
5.1 Introduction . . . 81
5.2 Motivations . . . 81
5.3 Prior Research . . . 82
5.3.1 Knowledge-Driven Approaches . . . 83
5.3.2 Data-Driven Approaches . . . 83
5.4 The Mutual Information OOV Model . . . 84
5.4.1 The Approach . . . 85
5.4.2 The Algorithm . . . 86
5.5 Experiments and Results . . . 88
5.5.1 Learning the Subword Inventory . . . 88
5.5.2 Detection Results . . . 92
5.5.3 Recognition Performance . . . 94
5.5.4 Phonetic Accuracy . . . 95
5.6 Summary . . . 96

6 Multi-Class Modelling for OOV Recognition **99**
6.1 Introduction . . . 99
6.2 Motivations . . . 99
6.3 Prior Research . . . 100
6.3.1 Part-of-Speech Approaches . . . 101
6.3.2 Word Clustering Approaches . . . 102
6.4 Approach . . . 103
6.4.1 Part-Of-Speech OOV Classes . . . 104
6.4.2 Automatically-Derived OOV Classes . . . 105
6.4.3 The Combined Approach . . . 107
6.5 Experiments and Results . . . 107
6.5.1 The POS Model . . . 107
6.5.2 The Automatically-Derived Model . . . 111
6.6 Summary . . . 115

7 Combining OOV Modelling and Confidence Scoring **117**
7.1 Introduction . . . 117
7.2 Prior Research . . . 118
7.3 Confidence Scoring in SUMMIT . . . 119
7.4 Combining OOV Detection and Confidence Scoring . . . 121
7.4.1 Approach . . . 121
7.5 Experiments and Results . . . 122
7.5.1 Experimental Setup . . . 122
7.5.2 Detecting OOV Words . . . 123
7.5.3 Detecting Recognition Errors . . . 123
7.5.4 The Combined Approach . . . 125
7.6 Summary . . . 127

8 Summary and Future Directions **129**
8.1 Summary . . . 129
8.2 Future Work . . . 134

A Initial Phone Inventory **139**

B Pairs and Mutual Information Scores **141**

C Utterance Level Confidence Features **145**

Bibliography **147**

List of Figures

1-1 Word and sentence error rates for IV and OOV utterances. 19

3-1 Vocabulary growth for nine corpora described in [Hetherington 1994] : vocabulary size, or number of unique words is plotted versus the corpus size. . 40
3-2 Vocabulary growth for JUPITER and HUB4. Vocabulary size, or number of unique words is plotted versus the corpus size. 41
3-3 Distribution for the number of phones per word for IV and OOV words. The distributions are not weighted by word frequency. The average IV word length is 5.37 phones, and the average OOV word length is 5.42 phones. . . 44
3-4 Distribution for the number of phones per word for IV and OOV words. The distributions are weighted by word frequency. The average IV word length is 3.64 phones, and the average OOV word length is 5.27 phones. 45

4-1 The proposed framework. 48
4-2 The IV search network. Only a finite set of words is allowed. Any sequence of the words can be generated during recognition. 50
4-3 An OOV search network that is based on subword units. Any unit sequence is allowed providing for the generation of all possible OOV words. 51
4-4 The hybrid search network. During recognition, the IV and OOV branches are explored at the same time to allow for OOV recognition 52
4-5 An FST T_u that enforces a minimum of $n = 3$ phones for an OOV word. All words with less than 3 phones are prohibited. 58
4-6 An FST T_u that allows for words between 3 and 5 phones long. All words less than 3 phones or more than 5 phones in length are prohibited. 59
4-7 Sample lexicon FST L. Only the two words $w_1 = aa$ and $w_2 = ab$ are in the vocabulary. 60
4-8 The complement lexicon FST $\bar{L}$. This FST allows for all phone sequences except aa and ab. 61
4-9 The shift s is the difference between the true start (or end) of a word and the hypothesized start (or end) of the word. 63
4-10 ROC curves for the three models: corpus, dictionary, and oracle. Each curve shows the DR versus the FAR for each of the models. 67
4-11 WER on the IV test set. Shown is the closed-vocabulary performance (10.9%) and the performance of the dictionary model system as a function of the FAR. 69
4-12 WER on the complete test set. Shown is the closed-vocabulary performance (17.1%) and the performance of the dictionary model system as a function of the FAR. 70

4-13 A breakdown of the system's performance in locating OOV words. The first plot shows, as function of the shift (or tolerance) s, the fraction of words with start *or* end aligning with a true boundary. The second plot shows the fraction of words where *both* start and end align with true boundaries. The third plot shows the fraction of words where both start and end align with true boundaries *and* of the OOV word. The last plot (dotted) shows the ratio of the third to the second plot. 72
4-14 Histograms for start (top) and end (bottom) shifts of correctly detected OOV words. 73
4-15 Cumulative distribution function of the shift for starts and ends of correctly detected OOV words. 74
4-16 ROC curve for OOV detection on JUPITER and HUB4. Both experiments use the same dictionary model to handle OOV words. 77
4-17 WER on the HUB4 test set. Shown is the closed-vocabulary performance (24.9%) and the performance of the dictionary model system as a function of the FAR. 78

5-1 The algorithm for learning variable-length multi-phoneme subword units. These units are used to construct the OOV model. 87
5-2 Ordered MI values for the first 20 iterations. For each iteration, all pairs of units are sorted in descending based on their weighted mutual information. The weighted mutual information is plotted as a function of the rank of the pair. 89
5-3 Distribution of unit length (in terms number of phonemes). The mean of the distribution is 3.2 phonemes and with a minimum of 1 and a maximum of 9. 93
5-4 Distribution of syllable length (in terms number of phonemes). The mean of the distribution is 3.9 phonemes and with a minimum of 1 and a maximum of 8. 93
5-5 ROC curves for the four models: corpus, oracle, dictionary, and MI. Each curve shows the DR versus the FAR for each of the models. 94
5-6 WER on the complete test set. Shown is the closed-vocabulary performance (17.1%) and the performance of the dictionary and MI models as a function of the FAR. 96

6-1 ROC plot for the POS multi-class model. Also provided the ROC for the baseline system of a single class model. The ROC shows the case where both the OOV network and the language model are multi-class models. 110
6-2 Weighted average perplexity of the multi-class model in terms of the clustering iteration number. Shown are the cases for an agglomerative clustering starting initialization and a POS tag initialization. 112
6-3 ROC plot for the two automatic multi-class models. Also provided the ROC for the baseline system of a single class model. Plots for both agglomerative clustering and POS initializations are shown. 113
6-4 ROC plots for the three systems: dictionary, mutual information, and automatic multi-class. 114
6-5 The FOM performance of the automatic model as a function of the number of classes. 114

7-1 Comparison of the rejection rate of errors caused by OOV words versus the false rejection rate of correctly recognized words. 124
7-2 ROC curves for OOV word detection and confidence scoring methods evaluated on all words and on keywords only. 125
7-3 ROC curves on hypothesized keywords only using the OOV word detection and confidence scoring methods as well as a combined approach. 126

List of Tables

2-1 Example of a call to JUPITER. 30
2-2 The seven focus conditions for the HUB4 broadcast news corpus. 31
2-3 Example from the HUB4 corpus for the F0 condition. 32

3-1 Top ten most frequent OOV words in JUPITER. The second column shows the number of times each word occurs and the third column gives a sample utterance for each word. 42
3-2 Top ten most frequent OOV words in HUB4. The second column shows the number of times each word occurs and the third column gives a sample utterance for each word. 43
3-3 Average number of phones per word for IV and OOV words. The second column shows the average not weighted by the frequency, the third column shows the averages weighted by the word frequency 43
3-4 Distribution of OOV words in the JUPITER and HUB4 domains. Each column shows the percentage of OOV words corresponding to the five types of OOV words. 44

4-1 Rates of substitution, insertion, and deletion and word error rates (WER) obtained with the JUPITER baseline recognizer on the complete test set (2,029 utterances) and on the IV portion of the test set (1,715 utterances). 64
4-2 Rates of substitution, insertion, and deletion and word error rates (WER) obtained with the HUB4 baseline recognizer on the F0 condition. 65
4-3 The figure of merit performance of various OOV models. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM). 68
4-4 Rates of substitution, insertion, and deletion and phonetic error rates (PER) obtained with the dictionary model for the three shift tolerance windows of 25, 50, and 100 msec. 73
4-5 The figure of merit performance of various minimum length constraints. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM). The baseline is the dictionary model. 75
4-6 The figure of merit performance of the complement OOV model. This model is obtained by creating a search network that prohibits IV words. 76

5-1 The top 10 pairs for the first iteration. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information. 90

5-2 The top 10 pairs after iteration 50. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information. 90
5-3 Model perplexity for three configurations: the phoneme model, the MI model of 1,977 units, and the complete vocabulary model of 99,202 units. 91
5-4 Sample pronunciations with merged units. Some of the units are legal English syllables such as *y_uw*. Others are either syllable fragments such as the phoneme *f* in the word *festival* or multi-syllable units such as *sh_ax_n_ax_l* in the word *unconditional*. 92
5-5 Breakdown of the learned units into four types: legal English syllables, one-vowel units, multi-vowel units, and consonant cluster units. 92
5-6 The figure of merit performance of the four OOV models. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM). 95
5-7 Rates of substitution, insertion, and deletion and phonetic error rates (PER) obtained with dictionary model and the MI model for a tolerance shift windows of 25 msec. 95

6-1 Word-level test set perplexity for OOV and IV test sets. The table shows the perplexity using a single class of OOV words, as well as for using the eight POS classes described above. 108
6-2 Phone-level model perplexity for the baseline single class OOV model, the eight models of the eight classes, and the weighted average of the resulting multi-class OOV model. Also given is the count from PRONLEX of the number of words in each class. 109
6-3 Detection results on different configurations of the POS model. Results are shown in terms of the FOM measure for four different conditions the baseline single class model, and adding multiple classes both at the language model as well as at the OOV network level. 109
6-4 The figure of merit performance of all OOV models we explored. The ones in bold face are the multi-class FOMs. 111

7-1 An example utterance with an OOV word *Franklin*. Shown the input, output of the recognizer and the word-level confidence scores. 122

A-1 Label set used throughout this thesis. The first column shows the label, and the second gives an example word. 140

B-1 The top 100 pairs for the first iteration. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information. 142
B-2 The top 100 pairs after iteration 50. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information. 143

Chapter 1

Introduction

1.1 Introduction

Speech recognition is the process of mapping a spoken utterance into a sequence of words. A speech recognizer achieves this goal by searching for the most likely word string among all possible word strings in the language. Most conventional speech recognition systems represent this search space as a directed graph of phone-like units. These graphs are typically determined by the allowable pronunciations of a given word vocabulary, with word (and thus phone) sequences being prioritized by word-level constraints such as statistical language models or n-grams. This framework has proven to be very effective, since it combines multiple knowledge sources into a single search space, rather than decoupling the search into multiple stages, each with the potential to introduce errors. Although multi-stage searches have been explored, they typically all operate with the word as a basic unit.

Although this framework has worked extremely well, the use of the word as the main unit of representation has some difficulties in certain situations. One common and serious problem is that of out-of-vocabulary (OOV) words. For any reasonably-sized domain, it is essentially impossible to predefine a word vocabulary that covers all words that may be encountered during recognition. For example, in the JUPITER weather information system [Zue et al. 2000], the recognizer is constantly faced with OOV words spoken by users. Examples of such words are city names or concepts that are not in the vocabulary of the recognizer. OOV words are always encountered by speech recognizers no matter how large the vocabulary is, since the vocabulary of any language is constantly growing and changing. However, the magnitude of the problem depends mainly on two factors: the size of the word

vocabulary and the level of mismatch between training corpus used to construct the vocabulary and the speech it is used on. For large vocabularies (>64,000 words) with a reasonable match between the training and testing, the OOV rate could be as low as a fraction of a percent. However, for cases where the vocabulary is on the order of a few thousand words, or there is some mismatch between training and testing, the OOV rate could be as high as 3%. Although this percentage might sound small, OOV words tend to spread errors into neighboring words, dramatically degrading overall recognition performance. When faced with an OOV word, a recognizer may hypothesize a similar-sounding word or words from the vocabulary in its place, causing the neighboring words to be mis-recognized. A similar phenomenon to OOV words is that of partially spoken words, which are typically produced in more conversational or spontaneous speech applications. These phenomena also tend to produce errors since the recognizer matches the phonetic sequence with the best-fitting words in its active vocabulary. The problem of partially spoken words can be viewed as a special case of the OOV problem since the speech recognizer is faced with a portion of the spoken word that may represent a new or unseen phone sequence.

Figure 1-1 demonstrates the severity of the OOV problem in the JUPITER weather domain. The word error rate (WER) is three times higher for utterances with OOV words than it is for in-vocabulary (IV) utterances. The increase in WER for OOV utterances can be attributed to three factors. The first factor is the mis-recognition of OOV words since they are not in the vocabulary. The second factor is the mis-recognition of words neighboring OOV words. The third factor is the high correlation between out-of-domain utterances and OOV utterances; i.e. OOV utterances tend to be out-of-domain and are inherently harder to recognize. In the same figure, we show the sentence error rate (percentage of sentences with at least one recognition error). For OOV sentences, sentence error rate is always 100% since the OOV word will always be mis-recognized.

In this thesis, we tackle the OOV problem for medium and large vocabulary speech recognition systems. We propose a novel approach for handling OOV words within a single-stage recognition architecture. To achieve this goal, an explicit and detailed model of OOV words is constructed and then used to augment the closed-vocabulary search space of a standard speech recognizer. This OOV model achieves open-vocabulary recognition through the use of more flexible subword units that can be concatenated during recognition to form new phone sequences corresponding to potential new words. Examples of such units

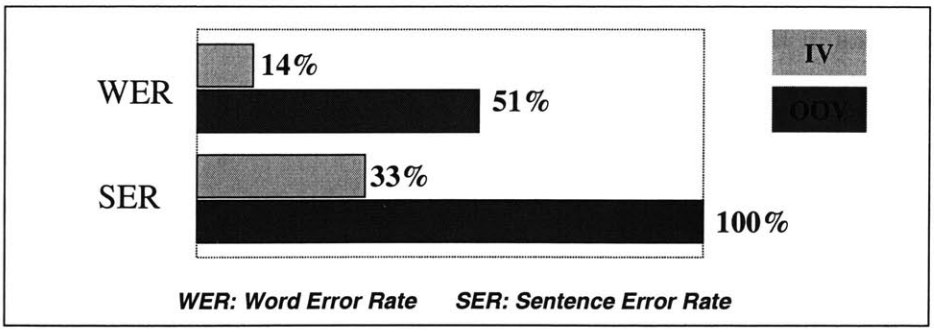

Figure 1-1: Word and sentence error rates for IV and OOV utterances.

are phones, syllables, or some automatically learned subword units. Subword units have the attractive property of being a closed set; thus they are able to cover any new words, and can conceivably cover most partial word utterances as well. The main challenge of such an approach is ensuring that the OOV model does not absorb portions of the speech signal corresponding to IV words. In dealing with this challenge, we explore several research issues related to designing the subword lexicon, language model, and topology of the OOV model. We present a dictionary-based approach for estimating subword language models. Such language models are utilized within the subword search space to help recognize the underlying phonetic transcription of OOV words. We also propose a data-driven iterative bottom-up procedure for automatically creating a multi-phone subword inventory. Starting with individual phones, this procedure uses the maximum mutual information principle to successively merge phones to obtain longer subword units. The thesis also extends this OOV approach to modelling multiple classes of OOV words. Instead of augmenting the word search space with a single model, we add several models, one for each class of words. We present two approaches for designing the OOV word classes. The first approach relies on using common part-of-speech (POS) tags. The second approach is a data-driven two-step clustering procedure, where the first step uses agglomerative clustering to derive an initial class assignment, while the second step uses iterative clustering to move words from one class to another in order to reduce the model perplexity.

The proposed recognition configuration can be used either to only handle OOV words in a speech recognition task or as a domain-independent first stage in a two-stage recognition system. With such a configuration we can separate domain-independent constraints

from domain-dependent ones in the speech recognition process while still utilizing word level constraints in the first stage. This is particularly useful in cases where the vocabulary of the system changes frequently because of some dynamic information source and can be incorporated into the second stage of the system. A two-stage recognizer configuration with OOV detection capability in the first stage might also provide for a more flexible deployment strategy. For example, a user interacting with several different spoken dialogue domains (e.g., weather, travel, entertainment) might have their speech initially processed by a domain-independent first stage, and then subsequently processed by domain-dependent recognizers. For client/server architectures, a two-stage recognition process could be configured to have the first stage run locally on small client devices (e.g., hand-held portables) and thus potentially require less bandwidth to communicate with remote servers for the second stage.

For spoken dialog systems, being able to detect the presence of an OOV word can significantly help in improving the quality of the dialog between the user and the system. In attempting to get the correct response from the system, the user may repeat an utterance with an OOV word several times, not knowing that the system can't recognize the utterance correctly. A system that can detect the presence of an OOV word may use a more effective dialog strategy to handle this situation.

1.2 Thesis Goals

There are four different problems that can be associated with OOV words. The first problem is that of detecting the presence of an OOV word. Given an utterance, the goal is to find out if it has any words that the recognizer does not have in its vocabulary. The second problem is to accurately locate the start and end of the OOV word within the utterance. The third problem is the recognition of the underlying sequence of subword units (e.g., phones) corresponding to the OOV word. The fourth problem is the identification problem: given a phonetic transcription of the word, the goal is to derive a spelling that best matches the recognized sequence of phones.

The primary goal of this thesis is to investigate and develop an approach for handling OOV words within a finite-state transducer recognition framework. The thesis focuses on the first three sub-problems of OOV recognition: detecting, locating, and phonetically

transcribing OOV words. The problem of identifying the spelling of the word is beyond the scope of this thesis. In achieving our primary goal, several research issues are addressed:

1. What are some of the most commonly used approaches to the OOV problem? What are the advantages and disadvantages of these approaches?

2. How can we enable a word-based recognizer to handle OOV words without compromising performance on IV words?

3. Can we simultaneously recognize IV words and subword units corresponding to OOV words within a single-stage recognition configuration? How can we construct such a recognizer?

4. What type of subword n-gram language models are most effective in predicting the OOV word structure?

5. What type of subword units should be used to construct a model for recognizing OOV words?

6. How can we combine the OOV approach with other techniques for robust speech recognition such as confidence scoring?

In this thesis, we make the following contributions to research in the area of out-of-vocabulary recognition:

- The development of a single-stage recognition framework for handling OOV words that combines word and subword recognition within the same search space.

- The development of dictionary-based techniques for training subword n-gram language models for use in OOV recognition.

- The development of a data-driven procedure for learning OOV multi-phone units based on the maximum mutual information principle.

- The development of a multi-class approach for modelling OOV word classes that is based on part-of-speech tags and a perplexity clustering procedure.

- Combining OOV modelling and confidence scoring to improve performance on the task of detecting recognition errors.

- Empirical studies demonstrating the applicability of our approach to medium and large vocabulary recognition tasks, and comparing various configurations of the OOV model.

1.3 Outline

The remainder of this thesis is organized into eight chapters. Following is a brief description of each chapter:

- **Chapter 2: Experimental Background**

  This chapter is intended to provide the basic background needed throughout the thesis. The chapter first describes the SUMMIT recognition system. A short overview of finite state transducers and their use for speech recognition is then given. The chapter also provides a short overview of the two corpora used in this thesis.

- **Chapter 3: Survey and Analysis**

  We present a survey of approaches to the OOV problem. The survey describes four main categories of approaches. The chapter then presents a brief analysis of vocabulary growth and OOV words for the two corpora we use in this thesis.

- **Chapter 4: A Single-Stage OOV Approach**

  This chapter describes our vision for open-vocabulary recognition. It also describes how we apply this framework for modelling OOV words within a single-stage recognition architecture. The chapter presents the layout as well as the probability model of the approach. Several configurations of the OOV model as well as various techniques to constrain OOV word recognition are presented in this chapter. The second half of the chapter presents a series of experiments on two domains. We describe the four performance measures relevant to the OOV problem including detection quality and recognition. We then describe the experimental setup of the two domains and report on a set of results. In addition, we discuss the impact of applying topology constraints to the model and the performance on large vocabulary domains.

- **Chapter 5: Learning Multi-Phone Units for OOV Recognition**

  This chapter presents an OOV model based on learning subword units using a mutual information criterion. We first review some related prior research. Next, we

describe a technique for learning an inventory of multi-phone variable-length units. The technique is an iterative bottom-up procedure that relies on a mutual information measure. A set of experiments on using this approach is then presented.

- **Chapter 6: A Multi-Class Extension**

  The chapter presents a multi-class extension to our approach for modelling OOV words. We present two methods for designing the OOV classes. The first is knowledge-driven and relies on using common POS tags to design the OOV classes. The second method is data-driven and uses a two-step clustering procedure. We also present an approach that combines these first two methods. Finally, we describe experimental results comparing the multi-class approaches to the baseline single-class approach.

- **Chapter 7: Combination with Confidence Scoring**

  We compare and contrast OOV modelling and confidence scoring in their ability to detect OOV words and recognition errors. We also present a method for combining the two techniques to detect recognition errors. Finally, we provide experimental results demonstrating the performance gains that can be obtained with the combined approach.

- **Chapter 8: Summary and Future Work**

  This chapter is a summary of the contributions and findings of the thesis. The chapter concludes with a discussion on possible future work.

Chapter 2

Experimental Setup

2.1 Introduction

This chapter is intended to provide the general background directly relevant to the content of this thesis. The chapter is divided into two parts. In the first part, we review the SUMMIT recognition system used for our empirical studies throughout the thesis. We, then, provide a short overview of finite-state transducers (FSTs) and how they are used within the SUMMIT system to represent the speech recognition problem. The chapter concludes with a description of the two corpora used in this thesis: the JUPITER weather information domain corpus and the HUB4 broadcast news domain corpus.

2.2 The SUMMIT Recognition System

All speech recognition work reported in this thesis is done within the MIT SUMMIT speech recognition system. Unlike most of the current frame-based speech systems, SUMMIT implements a segment-based speech recognition approach [Glass et al. 1996; Livescu 1999]. The system attempts to segment the waveform into predefined subword units, which may be phones, or other subword units. This is in contrast to frame-based recognizers, which divide the waveform into equal-length windows, or frames [Bahl et al. 1983; Rabiner 1989].

Mathematically, the speech recognition problem can be formulated as follows. For a given input waveform with corresponding acoustic feature vectors $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$, the goal is to find the most likely sequence of words $\mathbf{W}^* = \{w_1, w_2, \ldots, w_M\}$ that produced the waveform. This can be expressed as:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{A}), \tag{2.1}$$

where $\mathbf{W}$ ranges over all possible word sequences. This formulation is further expanded to account for various pronunciations $\mathbf{U}$ and different segmentations $\mathbf{S}$ as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_{\mathbf{U},\mathbf{S}} P(\mathbf{W}, \mathbf{U}, \mathbf{S}, |\mathbf{A}), \tag{2.2}$$

where $\mathbf{U}$ ranges over all possible pronunciations of $\mathbf{W}$ and $\mathbf{S}$ ranges over all possible segmentations for all of the pronunciations. Similar to most speech recognition systems, and to reduce the computational complexity of recognition, SUMMIT assumes that, given a word sequence $\mathbf{W}$, there is an optimal segmentation and unit sequence, which is much more likely than any other $\mathbf{S}$ and $\mathbf{U}$. Hence, the summation is approximated by a maximization, in which we attempt to find the best triple of word string, unit sequence, and segmentation, or the best path, given the acoustic features:

$$\{\mathbf{W}^*, \mathbf{U}^*, \mathbf{S}^*\} = \arg\max_{\mathbf{W},\mathbf{U},\mathbf{S}} P(\mathbf{W}, \mathbf{U}, \mathbf{S}, |\mathbf{A}) \tag{2.3}$$

When we apply Bayes' Rule, we obtain:

$$\{\mathbf{W}^*, \mathbf{U}^*, \mathbf{S}^*\} = \arg\max_{\mathbf{W},\mathbf{U},\mathbf{S}} \frac{P(\mathbf{A}|\mathbf{W}, \mathbf{U}, \mathbf{S})P(\mathbf{S}|\mathbf{U}, \mathbf{W})P(\mathbf{U}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{A})} \tag{2.4}$$

$$= \arg\max_{\mathbf{W},\mathbf{U},\mathbf{S}} P(\mathbf{A}|\mathbf{W}, \mathbf{U}, \mathbf{S})P(\mathbf{S}|\mathbf{U}, \mathbf{W})P(\mathbf{U}|\mathbf{W})P(\mathbf{W}) \tag{2.5}$$

The obtained formulation of the speech recognition problem is realized through Viterbi decoding [Bahl et al. 1983], and is identical for frame-based and segment-based recognizers. The difference lies in the fact that segment-based recognizers explicitly consider segment start and end times during the search, whereas frame-based methods do not. In the following sections we look in more details at each component of Equation 2.5. The estimation of the four components of Equation 2.5 is performed by, respectively, the acoustic, duration, lexical (or pronunciation), and language model components. For our work in this thesis, $P(\mathbf{S}|\mathbf{U}, \mathbf{W})$ is constant, and hence no duration model is used.

2.2.1 Segmentation

As a segment-based recognition system, SUMMIT starts by transforming the input waveform into a network of possible segmentations. This is done by first extracting frame-based acoustic features (*e.g.*, MFCC's) at equal intervals, as in a frame-based recognizer, and then hypothesizing groupings of frames which define possible segments. There are various methods for performing this task; two that have been used in SUMMIT are acoustic segmentation [Glass 1988; Lee 1998], in which segment boundaries are hypothesized at points of large acoustic change, and probabilistic segmentation [Chang 1998; Lee 1998], which uses a frame-based phonetic recognizer to hypothesize boundaries. Acoustic segmentation is used throughout this thesis.

2.2.2 Acoustic Modelling

Once a segmentation graph is hypothesized, a vector of acoustic features is extracted for each segment or boundary in the segmentation graph, or for both the segments and boundaries in the graph. In this thesis, we use only the boundaries in the segment graph. Each hypothesized boundary in the graph may be an actual transition boundary between subword units, or an internal boundary within a phonetic unit. We refer to both of these as boundaries or diphones, although the latter kind does not correspond to an actual boundary between a pair of phones. The acoustic features are now represented as a set of boundary feature vectors. SUMMIT assumes that boundaries are independent of each other and of the word sequence as well of the pronunciation of the words in the word sequence. During training, the feature vectors are used to train diphone acoustic models. During recognition, the feature vectors are scored against the available acoustic models to determine the most likely word sequence, during Viterbi decoding.

2.2.3 Lexical Modelling

Knowledge about the allowable set of words, and their respective pronunciations is represented through the lexical model, or simply the lexicon. The lexicon is simply a dictionary of allowable pronunciations for all of the words in the recognizer's vocabulary [Zue et al. 1990]. The lexicon consists of one or more basic pronunciations, also referred to as baseforms, for each word, as well as any number of alternate pronunciations created by applying

phonological rules to the baseform. The phonological rules account for processes such as place assimilation, gemination, and alveolar stop flapping [Hetherington 2001]. The alternate pronunciations are represented as a graph. This pronunciation graph can be weighted to account for the likelihood of various pronunciation. For this thesis no pronunciation weights are used.

2.2.4 Language Modelling

The type of language model used in this thesis is the standard statistical n-gram [Bahl et al. 1983]. The n-gram provides an estimate of $P(\mathbf{W})$, the probability of observing the word sequence $\mathbf{W}$. Assuming that the probability of a given word depends on a finite number $n-1$ of preceding words, The probability of an N-word string can be written as:

$$P(\mathbf{W}) = \prod_{i=1}^{N} P(w_i | w_{i-1}, w_{i-2}, \ldots, w_{i-(n-1)}) \tag{2.6}$$

N-gram language models are typically trained by counting the number of times each n-word sequence occurs in a set of training data. Smoothing methods are often used to redistribute some of the probability from observed n-grams to unobserved n-grams [Chen and Goodman 1998]. In this thesis, we use word, phone, as well as multi-phone n-grams with $n = 2$ or $n = 3$. For this thesis, an expectation maximization smoothing is used to estimate unseen n-gram probabilities [Dempster et al. 1977].

2.2.5 Recognition

The goal of the recognizer is to find the best-scoring path through the recognition search space. The search space is the aggregation of all components we presented in the previous subsections.

The search space is created by combining the scored segmentation graph, the lexicon, and the language model. The language model is applied in two passes. In the first pass, a forward Viterbi beam search [Rabiner and Juang 1993] is used to obtain the partial-path scores to each node using a bigram language model, followed by a backward A^* beam search [Winston 1992] using the scores from the Viterbi search as the look-ahead function. The A^* search produces a smaller word graph, which can then be searched with a more detailed language model such as a trigram. In the second pass, the new language model is

used to rescore the word graph with forward partial path scores, and a backward A^* search is again used to obtain the final hypotheses.

2.3 Finite-State Transducers

The underlying implementation of the SUMMIT system is based on a weighted finite-state transducer (FST) framework. The framework allows for a uniform representation of the information sources used in recognition, including context-dependent units, pronunciation dictionaries, language models and lattices. Furthermore, general but efficient algorithms can be used for combining information sources in actual recognizers and for optimizing their application [Mohri and Riley 1997]. Within the FST framework, an FST recognizer R can be represented as the composition of four FSTs [Glass et al. 1999],

$$R = C \circ P \circ L \circ G, \tag{2.7}$$

where C represents the mapping from context-dependent to context-independent phonetic units; P represents the phonological rules that map phonemes to phones (or visa versa); L represents the word lexicon; and G represents the language model. The process of recognition then becomes a search for the best path through R. SUMMIT allows the composition of P, L, and G to be performed either during or prior to recognition.

Any of these transductions can be weighted. A big advantage of this formulation is that the search components operate on a *single* transducer R; the details of its composition are not a concern to the search. As such, R can be precomputed and optimized in various ways or it can be computed on demand as needed. This use of a cascade of weighted finite-state transducers is heavily inspired by work at AT&T [Mohri and Riley 1997].

2.4 The Corpora

Two corpora are used for our experimental studies throughout this thesis: the JUPITER weather information domain corpus and the HUB4 broadcast news domain corpus. The next two sections describe each corpus in more detail.

Caller: what is the forecast for today in Boston.
JUPITER: in Boston today: mostly sunny. hot and humid. high lower 90s. tonight: mostly clear and muggy. lows lower 70s in the city and 65 to 70 elsewhere. What other information can I give you?
Caller: what is the forecast for today in California
JUPITER: I know of the following places in California: Burbank, Fresno, Long Beach, Los Angeles, Menlo Park, Monterey, Oakland, Palo Alto, Pasadena, Sacramento, San Diego, San Francisco, San Jose, Santa Barbara, and Santa Clara. Please select one.
Caller: Sana Barbara please.
JUPITER: in Santa Barbara today: high 73. tonight. mostly cloudy conditions expected through the night. low 58 tonight. Is there anything else?
Caller: Nothing thank you.
JUPITER: Thanks for using JUPITER! Have a nice day!

Table 2-1: Example of a call to JUPITER.

2.4.1 The JUPITER Corpus

JUPITER is a telephone-based conversational system that provides weather information for over 600 cities, both in the United States as well as worldwide [Zue et al. 2000; Zue et al. 1997; Glass and Hazen 1998]. Using a toll-free number, users can call JUPITER and inquire about various aspects of current weather forecasts, such as temperature, humidity, wind speed, precipitation, sunrise and sunset times, and weather advisories. The system includes a complete dialog management component, where users can ask JUPITER for help or clarification in various ways, such as "What cities do you know in California?" or "Is it going to snow tomorrow?" During a call, JUPITER keeps a short history of dialogue-specific information, so that questions such as "What about tomorrow?" can be answered appropriately. References are resolved using a context resolution mechanism based on the state of the dialog from the current and previous queries. Table 2-1 shows an example of a call to JUPITER.

The corpus is created by recording utterances from calls made to JUPITER. The corpus used in this thesis contains 88,755 utterances for training collected between February 1997 and July 1999. Most of the utterances were collected live via the toll-free number. The corpus also contains approximately 3,500 read utterances, as well as about 1,000 utterances obtained through wizard data collection, in which the input is spontaneous but a human typist replaces the computer [Glass and Hazen 1998]. In our experiments, we used only the

live data for training and testing. For each utterance, the corpus contains an orthographic transcription produced by a human transcriber.

2.4.2 The HUB4 Corpus

A complete description of the HUB4 corpus is provided in [Graff and Liberman 1997]. Here we provide a short description of the domain and the corpus. The HUB4 domain corpus contains recorded audio from various US broadcast news shows, including both television and radio. The total amount of data is 106 hours of recorded audio referred to as *BNtrain96* (34 hours) and *BNtrain97* (72 hours). The 106 hours of audio were annotated to include a speaker identification key, background noise condition, and channel characteristics. The corpus contains a variety of recording conditions. Table 2-2 shows the various conditions under which the data was collected [Kubala et al. 1997].

Focus	**Description**
F0	clean planned speech
F1	spontaneous clean broadcast speech
F2	low fidelity speech, narrowband
F3	speech with background music
F4	degraded acoustic conditions
F5	non-native speakers, clean and planned
FX	all other speech, combining various conditions

Table 2-2: The seven focus conditions for the HUB4 broadcast news corpus.

Table 2-3 shows an example taken from the F0 condition. Each segment is annotated with the start and end times, as well as the type and identification of the speaker.

The training data for the language model consists of significantly larger amount of data, up to about 200 million words, collected from various sources including the Wall Street Journal, as well as various transcriptions from televisions and radio shows on stations such as CNN and NPR [Kubala et al. 1997].

2.5 Summary

In this chapter we briefly covered background relevant to this thesis. We first described the SUMMIT speech recognition system that is used for all our experimental studies throughout

Annotated example from HUB4
<section type=report startTime=28.160 endTime=194.311>
<turn speaker=spkr1 spkrtype=male startTime=28.160 endTime=32.160>
<time sec=28.160>
"From A B C, this is World News Tonight with Peter Jennings"
< /turn>
<turn speaker=Peter Jennings spkrtype=male startTime=32.160 endTime=60.580>
<time sec=32.160>
"Good evening. The jury couldn't agree, and ..."
<time sec=50.017>

Table 2-3: Example from the HUB4 corpus for the F0 condition.

this thesis. We described each of the components of the system including the acoustic and language model elements as well as the training and recognition procedure. Then, we described the FST representation of the recognition framework. Finally, we presented a description of the two corpora used for our experimental studies in this thesis.

Chapter 3

Survey and Analysis

3.1 Introduction

In this chapter, we present a survey on the OOV problem in continuous speech recognition. We start by describing four main categories of approaches to handling OOV words. We then highlight some of the previous work in the field. We conclude with a brief analysis of vocabulary growth and OOV words in the two corpora we use in this thesis.

3.2 Approaches

Approaches to the OOV problem can be classified into four categories. Here we only describe the four approaches to give a broad view of prior research on the problem. The following section provides concrete examples from the literature describing various previous attempts at these approaches.

3.2.1 Vocabulary Optimization

The first strategy is vocabulary optimization. By vocabulary optimization we mean designing the vocabulary in such a way to reduce the OOV rate by as much as possible. Vocabulary optimization could either involve increasing the vocabulary size for large vocabulary domain-independent recognizers, or it could also be selecting those words that are most frequent in the specific domain of a domain-dependent recognizer. In either case, the OOV problem can not be totally eliminated because words form an open set and there will always be new words encountered during recognition. A good example is proper names,

as well as foreign words. Another drawback of the vocabulary optimization approach is that increasing the vocabulary size makes the recognition more computationally expensive and could degrade performance due to the larger number of words to choose from during recognition.

3.2.2 Confidence Scoring

The second strategy is the use of confidence scoring to predict whether a recognized word is actually a substitution of an OOV word. Using confidence scoring is an example of an implicit approach at solving the problem, and is done by observing some parameters from the recognition engine. Examples of confidence measures are acoustic scores, statistics derived from the language model, and statistics derived from an N-best sentence list of recognition. The ability to estimate the confidence of the hypothesis allows the recognizer to either reject all or part of the utterance if the confidence is below some threshold. The main weakness of this strategy is that such confidence measures are good at predicting whether a hypothesized word is correct or not, but unable to tease apart errors due to OOV words from those errors due to other phenomena such as degraded acoustic conditions.

3.2.3 Multi-Stage Subword Recognition

The third strategy is the multi-stage recognition approach. This approach involves breaking the recognition process into two or more stages. In the early stage(s), a subword recognition is performed to obtain phonetic sequences that may or may not be in the recognizer's word vocabulary. By performing the recognition in two steps, the recognizer gets the chance to hypothesize novel phonetic sequences. These are phone sequences that could potentially correspond to an OOV word. The second stage involves mapping the subword output of the first stage into word sequences using word-level constraints.

There are many variations to the approach. For examples, the first stage can either be a phonetic-level recognizer, a syllable-level recognizer, or some automatically-derived subword units. In addition, the output of the first stage can either be the top hypothesis from the recognizer, the top N hypotheses, or a graph representing a pruned search space of the first stage.

The main drawback of the multi-stage approach is the fact the an important piece of knowledge, the word-level lexical knowledge, is not utilized early enough. Instead, it is

delayed to some later stage causing performance on IV words to degrade. The degradation can be partially avoided by increasing the size of the graph from the first stage to preserve a significant portion of the complete search space. The limitation is the computation and storage needed to manipulate the graph in the later stage(s).

3.2.4 Filler Models

Filler models are by far the most commonly used approach for handling OOV words. The approach involves adding a special lexical entry to the recognizer's vocabulary that represents an OOV word. A filler model typically acts as a generic word or a garbage model. The model competes with models of in-vocabulary words during recognition and a hypothesis including a path through the filler model signals the presence of an OOV word. There are several variations on the structure of a filler model, some of which include imposing constraints on the sequence of phones allowed. In addition, most of the filler models used rely on less-detailed acoustic models than those used for in-vocabulary words. Another variation is the language model component and how the transitions into and out of the filler model are controlled during recognition.

We should mention here that the notion of filler models is also common in keyword spotting [Manos and Zue 1997; Manos 1996], where the filler is used to absorb those words that are of no interest to the keyword spotter. The main distinction between OOV modelling and keyword spotting is that in keyword spotting, the intention is to absorb those *unimportant* words with the filler, while in OOV modelling, the filler is used to detect and hence absorb OOV words which are possibly the most important in the utterance.

The main drawback of filler models is the fact that they are highly unconstrained and can potentially absorb parts of the speech signal corresponding to in-vocabulary words.

3.3 Prior Research

We start with some of the earliest work in the field [Asadi et al. 1991; Asadi 1991; Asadi and Leung 1993]. Asadi *et al.* explored both the OOV detection problem as well as the OOV acquisition or learning problem. Their approach used the filler model idea and they experimented with various configurations of the filler model. They reported results on the Resource Management (RM) task [Price et al. 1988], using the BBN BYBLOS continuous

speech recognition system [Chow et al. 1990]. The BYBLOS system used HMMs and a statistical class bigram language model. The utterances in the RM task were generated artificially from a finite-state grammar, so there were no true OOV words. They artificially created 55 OOV words by removing those words from the standard RM lexicon. Most of the simulated OOV words were proper nouns. The filler models they reported were networks of all phone models with an enforced minimum number of phonemes (2–4). Both context-independent and context-dependent phonetic models were examined. Because of the constrained nature of the task, they were able to allow OOV words in only appropriate locations within the utterance, making the OOV problem artificially easier. Overall, Asadi *et al.* found that an acoustic model requiring a sequence of at least two context-independent phonemes yielded the best detection result achieving 60–70% detection rate with a 2–6% false-alarm rate. Performance degraded when they went from context-independent to a more detailed context-dependent filler model. They attributed this to the fact that the system used context-dependent phoneme models for in-vocabulary words, and thus the filler model tended to mistakenly allow in-vocabulary words through the filler model. In effect, they found it advantageous to bias the system away from new words by using less-detailed acoustic models for them. Because of the artificial and the constrained nature of the RM task, it is not clear how well their results generalize to real, non-simulated OOV words. Nevertheless, they were the first to attempt the use of a filler-type model for handling OOV words. In addition to detecting OOV words, they experimented with various strategies for transcribing new words with the help of a sound-to-letter system, subsequently adding new words to the vocabulary and language model of the recognizer.

Jusek *et al.* [Jusek et al. 1995] created two German syllable models using phonotactic knowledge. The syllables were represented with a phone network which encoded allowed transitions between consonants or consonant clusters and vowels. The OOV model consisted of concatenating the phonotactic syllable networks. Compared to the approach reported by Asadi, this was a more constrained filler model where only specific phonetic sequences were allowed during recognition. Similar to Asadi's work, this filler model was added as a lexical entry and a context-independent acoustic models were used. Experiments were performed on the 1995 Evaluation set of VerbMobil corpus which is part of a speech-to-speech dialog system [Bub and Schwinn 1996; Bub et al. 1997]. The OOV model was able to detect only 25% of the OOV words, but the overall accuracy of the system decreased slightly compared

to a closed-vocabulary system. Later in [Kemp and Jusek 1996], Kemp *et al.* extended this approach to include transition probabilities within the phonotactic network. Instead of incorporating the filler model as a single lexicon entry, they treated each phoneme as a lexical entry, augmenting the language model with transition probabilities from phonemes to words, phonemes to phonemes, in addition to the standard word to word transition probabilities. The performance with this approach yielded a slight improvement in the accuracy of the baseline system, but was able to detect only 10% of all OOV words.

The two approaches we reviewed so far used very constrained testing conditions with grammars that allow OOV words only in particular slots in the sentence. The context of the OOV word is mostly ignored in relationship to other words in the utterance, and the task of detecting OOV words was artificially simplified. When dealing with less-constrained, or totally-unconstrained tasks, the problem becomes considerably more difficult, and a language model that can predict an OOV word becomes an essential part of the solution. Suhm *et al.* identified this issue and showed the importance of building an OOV-aware language model [Suhm et al. 1993]. They reported results on a conference registration task, where they explored both OOV detection and OOV phonetic transcription. They used an OOV model similar to that reported by Asadi [Asadi et al. 1991]. The test set consisted of 59 utterances containing 42 names. All names were removed from the vocabulary to simulate new words, thus leaving them with 42 occurrences of new words. For language modelling they mapped all words in the training corpus which were not included in the vocabulary into one symbol, in effect explicitly modelling the context of an OOV word. They achieved a detection rate of 70% for a false alarm rate of 2% under the artificial condition of removing all names from the vocabulary and trying to detect them using this approach. Even though their approach was the first to use a language model on the OOV word, the experiments were only on simulated new words within a very small-vocabulary read-speech corpus.

Most recently is the work by Chung [Chung 2001]. Chung adopted a multi-stage architecture for speech recognition that can detect and recognize OOV words. The system she used to recognize OOV words is a three-stage system that is designed to incorporate an extensive linguistic knowledge both at the subword-level [Seneff et al. 1996], as well as at the natural language level [Seneff 1992]. The goal of the first stage is to extract an optimized phonetic network whose arc weights are the acoustic scores and the language model scores.

The second stage traverses the pruned search space and identifies potential word hypothesis as well as possible locations of OOV words. The third stage performs natural language processing on an N-best list from the second stage to improve performance over the second stage. Experiments reported on this work are with the JUPITER weather information domain. The test set was chosen so that every utterance in the test set contained at least one OOV word. All OOV words were unknown city names. The three stage system reduced word error rate by as much as 29% and understanding error rate (UER) by 67%. Although those results are impressive, there are two important caveats: first, a correct detection of an OOV word was considered a correct recognition in computing WER and UER; second, performance was only measured on OOV data and was not reported on IV data where a degradation in performance is possible with the multi-stage approach. In terms of learning OOV words, the system had the capability of phonetically transcribing OOV words, as well as proposing a spelling for the purpose of incorporating new words into the recognizer.

In [Bazzi and Glass 2000a], we explored the use of the phone and syllable as primary units of representation in the first stage of a two-stage recognizer. A finite-state transducer speech recognizer is utilized to configure the recognition as a two-stage process, where either phone or syllable graphs are computed in the first stage, and passed to the second stage to determine the most likely word hypotheses. Experiments on the JUPITER weather information domain showed that the two-stage system, with phones in the first stage, degraded performance on IV utterances from 10.4% to 15.7%. With syllables in the first stages, the degradation was from 10.4% to 13.2%. These results demonstrated that if we were to totally remove word-level constraints from the first stage, IV utterances would suffer significant degradation in performance. Because IV utterances make up the majority of spoken utterances (80-90%), any gains from handling OOV utterances could not make up for the significant loss in performance on IV utterances.

Other approaches include resorting to utterance rejection techniques such as in [Chase 1997a; Schaaf and Kemp 1997] where a confidence score is used to reject utterances that could belong to OOV words. Other techniques involve [Hayamizu et al. 1993] using a word and a phone recognizer in parallel and comparing their relative score as a phone recognizer can perform better when an OOV word is encountered. An approach similar to the work by Chung [Chung 2001] is that reported by De Mori [De Mori and Galler 1996] where a multi-stage approach is adopted. In [Kita et al. 1991], Kita *et al.* explored new-word

detection and transcription in continuous Japanese speech. Itou *et al.* [Itou et al. 1992] also performed joint recognition and new-word transcription in a continuous Japanese speech recognition system. Jelinek *et al.* [Jelinek et al. 1990] studied the problem of incorporating a new word into a statistical word n-gram language model. Their approach was to assign words to word classes based to context. Boros *et al.* [Boros et al. 1997] explored issues in handling OOV words during semantic processing and dialog management as part of a spoken dialog system. Brill [Brill 1993] studied the problem of assigning part-of-speech to new words. This work was a component of a part-of-speech tagger used to tag large bodies of text automatically.

In summary, the prior work on the OOV problem varied from simple detection approaches where the primary goal is to predict whether an utterance contains an OOV word or not, to the more sophisticated techniques that tried to use rich linguistic sources to learn and identify the OOV words, in addition to detecting them.

3.4 Vocabulary Growth and Coverage

The presence of OOV words is closely related to how the vocabulary of the recognizer is chosen. There are two characteristics of the recognizer vocabulary that influence the magnitude of the OOV problem. The first is the growth of the vocabulary as a function of the size of the training corpus, and second is the coverage of that vocabulary on some unseen data. These two characteristics are highly correlated since a fast vocabulary growth could be an indication of a higher OOV rate, i.e. lower coverage on unseen data.

In [Hetherington 1994], an excellent and extensive study of vocabulary growth and coverage is presented. The study covers nine different corpora such as the Wall Street Journal (WSJ) corpus [Paul and Baker 1992] and the ATIS corpus [Polifroni et al. 1991]. Results from [Hetherington 1994] are shown in Figure 3-1. Hetherington found that, based on vocabulary growth, corpora can be grouped in three clusters of the following characteristics:

1. Human-human written communication corpora, such as WSJ shows the highest vocabulary growth and the lowest coverage (highest OOV rate).

2. Human-human spoken communication corpora, such as Switchboard [Godfrey et al. 1992]. These corpora show a medium vocabulary growth and a medium coverage.

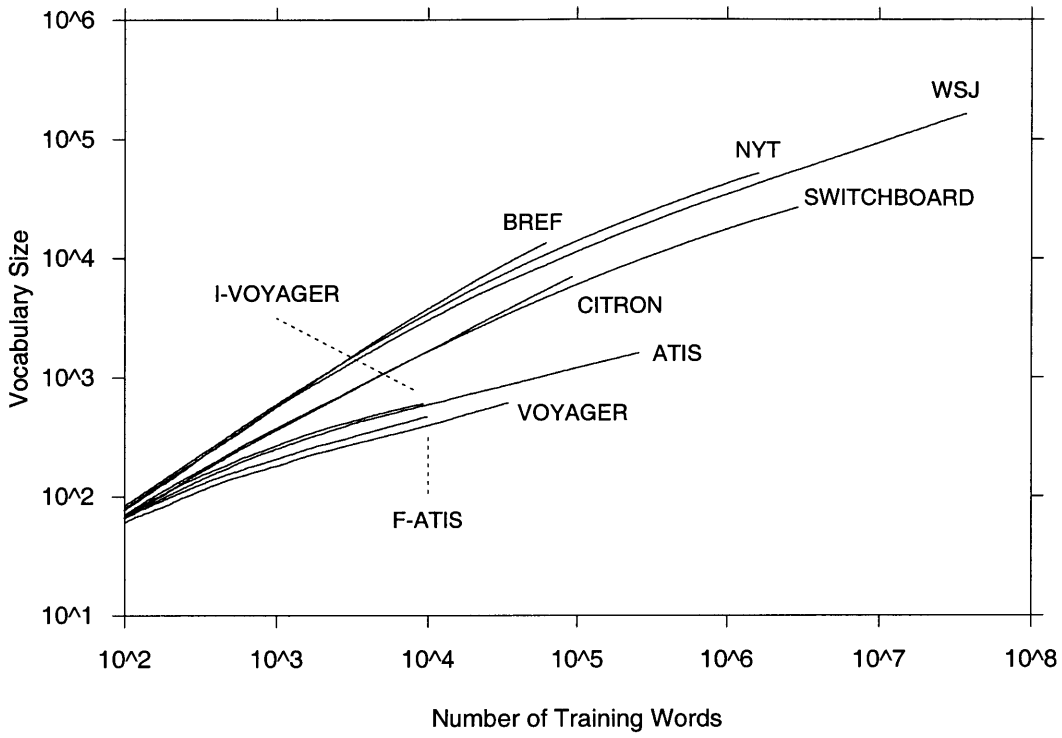

Figure 3-1: Vocabulary growth for nine corpora described in [Hetherington 1994] : vocabulary size, or number of unique words is plotted versus the corpus size.

3. Human-to-machine interaction corpora, or spoken dialog systems, such as ATIS. These corpora show a slow vocabulary growth and a moderate OOV rate.

Hetherington reported that a 1% OOV rate corresponds to around 17% of all sentences contain OOV words, both in the WSJ and ATIS corpora. Even though the OOV rates are domain-dependent, they are generally consistent across a wide range of applications. Another important finding by Hetherington is that even in systems with large vocabulary, over 100,000 words, the OOV rate could exceed 1%.

For the two domains we examine in this thesis, the vocabulary growth is shown in Figure 3-2. The plots show the relationship between corpus size and vocabulary size. For the JUPITER corpus, the vocabulary grows significantly slower than that of the HUB4 corpus. For around 0.5M words corpus size, the vocabulary size of JUPITER is only about 5,000 words, or one quarter of the 20,000 words vocabulary of HUB4 for the same corpus size. This behavior is consistent with Hetherington's findings that vocabulary growth depends on the application. The HUB4 domain falls under human-human interaction, written as well as spoken. The JUPITER domain falls under human-machine interaction and is of significantly

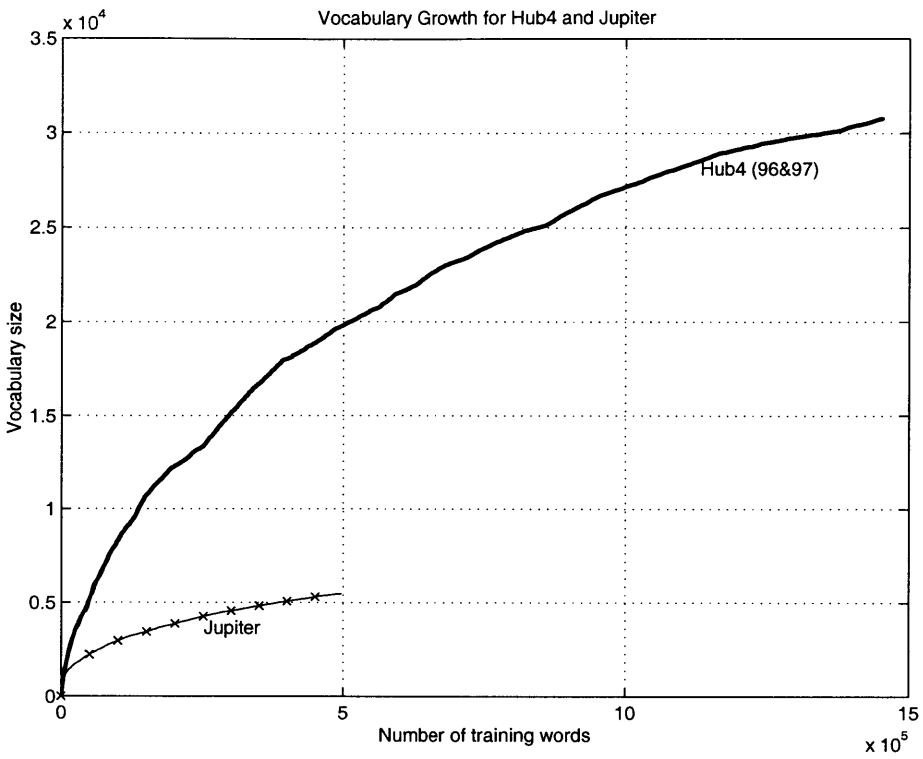

Figure 3-2: Vocabulary growth for JUPITER and HUB4. Vocabulary size, or number of unique words is plotted versus the corpus size.

slower vocabulary growth.

As to the OOV rate of our two domains, the JUPITER recognizer is setup to run with a carefully designed weather vocabulary that includes weather terms and city names. Under a vocabulary size of roughly 2,000 words, the OOV rate measured on unseen data averages 2.2%. For the HUB4 domain, the OOV rate ranges ranges from 1% up to 3% depending on the size and selection criterion of the vocabulary. For the recognition system we present later on in the following chapter, the OOV rate is 2.7% at a vocabulary size of 25,000 words.

3.5 Analysis of OOV Words

Hetherington [Hetherington 1994] provides a thorough study of the properties of OOV words including classifying new words into various part-of-speech (POS) classes, their phonological properties, as well as OOV word length. In this section, we look at some characteristics of OOV words in the JUPITER and HUB4 domains.

First, we examine the most common OOV words in the two domains. The top ten most

OOV word	Count	Sample Utterance
Clinton	16	what did Bill *Clinton* do today in the news
Bonnie	14	please tell me about the conditions of hurricane *Bonnie*
house	14	is it raining over the white *house*
problem	14	is there any place that's going to have a snow *problem*
Auburn	13	how windy is it in *Auburn* tomorrow
Belleville	13	weather *Belleville* Kansas this evening
blue	13	why is the sky *blue*
curious	13	sorry i was just *curious*
Dubai[1]	13	is *Dubai* expecting rain Tuesday
hell	13	what the *hell* is wrong with you

Table 3-1: Top ten most frequent OOV words in JUPITER. The second column shows the number of times each word occurs and the third column gives a sample utterance for each word.

frequent OOV words encountered in the JUPITER corpus are shown in Table 3-1. Similarly, Table 3-2 shows the top ten most frequent OOV words for the F0 condition of HUB4.

Examining some of the most frequent OOV words in JUPITER, we can classify these words into two categories. The first category is that of out-of-domain (OOD) words. Examples are the words *Clinton* and *house* that are not related to the weather domain. The second category is that of words that *do* belong to the domain but were left out because they were either not encountered in the training data or excluded from the lexicon because they were not frequent enough. Examples from Table 3-1 are the city names *Auburn* and *Dubai*[1]. For the HUB4 domain, OOV words tend to be either general English words that were not encountered in the training set such as *trainers* and *musicians* in Table 3-2, or proper names related to stories in broadcast news such as *Texaco* and *Gammage*.

3.5.1 Length of OOV words

One property we are interested in is the length of OOV words and how it compares to the length of IV words. Figure 3-3 shows the length distribution of IV and OOV words in the JUPITER corpus. The length is simply the number of phones of each word. The main observation from the distributions is that OOV and IV words have similar length distributions. The average length of an OOV word, shown in the second column Table 3-3,

[1]The word *Dubai* was originally in the vocabulary, but was removed because it was constantly confused with the word *good-bye*. *Good-bye* is frequently used in the JUPITER system.

OOV word	Count	Sample Utterance
Texaco	8	Jesse Jackson meets today with *Texaco* executives
Gammage	7	motorist Jonny *Gammage* of Syracuse New York defense team
Chatters	5	that is because *Chatters* believes this man
musicians	5	the *musicians* say they oppose this offer
Vojtas	5	considering charges against John *Vojtas*
Minthorn	4	but *Minthorn* says they agree
trainers	4	charges have been filed against three military *trainers*
Carlsbad	3	the *Carlsbad* fire has destroyed more than sixty homes
Pennington	3	*Pennington* says federal grants have helped him
tribes	3	northwest *tribes* have differing views

Table 3-2: Top ten most frequent OOV words in HUB4. The second column shows the number of times each word occurs and the third column gives a sample utterance for each word.

is 5.41 phones, slightly longer than the length of IV words (5.37 phones).

A different approach to looking at the length of OOV words is to examine their usage within the corpus. Figure 3-4 shows the length of IV and OOV words, weighted by how often these words occur in the JUPITER corpus. The main observation from these distributions is that IV words usage is dominated by short words, typically function words like *if* and *the*. On the other hand, OOV length distribution does not change much when weighted by the frequency of usage. For the frequency-weighted case, the average length of an OOV word is 5.27 phones, 45% longer than that of IV words.

Type	Not Weighted	Weighted
JUPITER IV words	5.37	3.64
JUPITER OOV words	5.41	5.27
HUB4 IV words	6.29	4.32
HUB4 OOV words	6.93	6.84

Table 3-3: Average number of phones per word for IV and OOV words. The second column shows the average not weighted by the frequency, the third column shows the averages weighted by the word frequency

Table 3-3 also shows average word length for the HUB4 domain. The average length of both IV and OOV words is longer than those on the JUPITER domain. Furthermore, OOV words are longer than IV words even when unweighted by the frequency of usage. As shown in Table 3-3, OOV words are on average 6.93 phones long. When weighted by the frequency

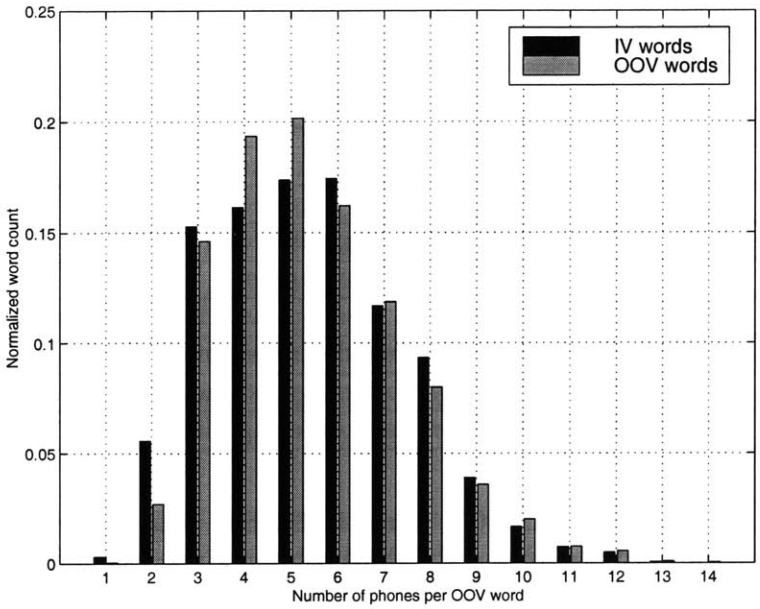

Figure 3-3: Distribution for the number of phones per word for IV and OOV words. The distributions are not weighted by word frequency. The average IV word length is 5.37 phones, and the average OOV word length is 5.42 phones.

of usage, OOV words are on average 6.84 phones long. The distributions of the word length in the HUB4 domain are similar to those on JUPITER but with higher averages.

Type	JUPITER (%)	HUB4 (%)
NOUN	41.0	29.1
NAME	35.2	48.3
VERB	15.3	11.1
ADJECTIVE	6.2	8.7
ADVERB	2.3	2.8

Table 3-4: Distribution of OOV words in the JUPITER and HUB4 domains. Each column shows the percentage of OOV words corresponding to the five types of OOV words.

3.5.2 Types of OOV Words

In Table 3-4, we break down OOV words into five classes. One class is that of proper names, while the other four correspond to syntactic part-of-speech (POS) tags. For JUPITER, nouns make up for the largest portion of 41.0%. Examples of nouns include words such as *desert, miles,* and *house.* The second class is proper names and makes up for 35.2% of OOV words. For the JUPITER domain, proper names are mainly city names, countries, and

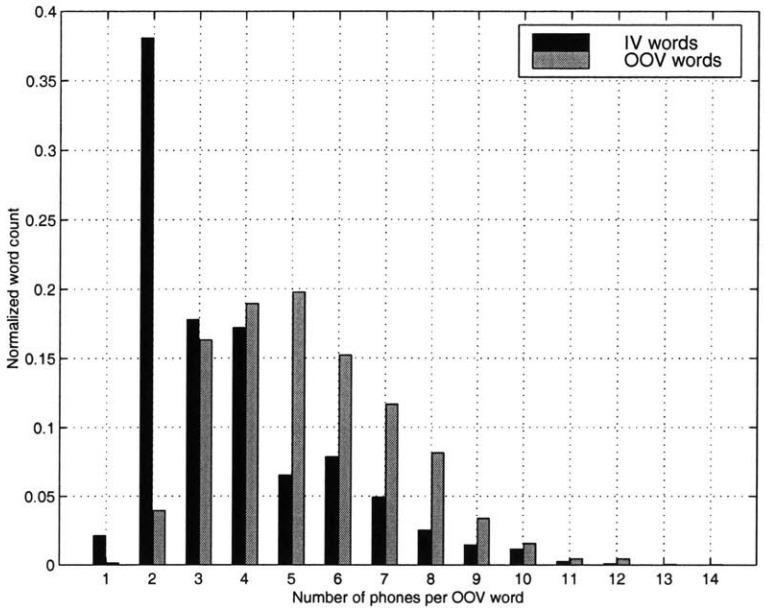

Figure 3-4: Distribution for the number of phones per word for IV and OOV words. The distributions are weighted by word frequency. The average IV word length is 3.64 phones, and the average OOV word length is 5.27 phones.

places. Examples include names such as *Beaverton, Franklin*, and *Bonnie*. The last three categories are verbs, adjectives, and adverbs. They make up for less than one third of occurring OOV words. Other types of words, such as function words are absent from the list shown in Table 3-4. This is because all function words are already in the vocabulary.

For HUB4, the distribution is different. Proper names make almost half of the OOV words at 48.2%. Nouns come next at 29.1% and then verbs, adjectives and adverbs. The difference in the distribution is mainly due to the fact that with large vocabulary, many of the nouns and verbs become part of the vocabulary. So, names are the most likely OOV words.

3.6 Summary

In this chapter we presented a survey of the OOV problem and a brief analysis of vocabulary growth and OOV words. We first categorized approaches to the OOV problem into four distinct strategies. These strategies are vocabulary optimization, confidence scoring, multi-stage recognition, and filler models. We, then, reviewed some of the prior research in the field. Next, we presented an analysis study of vocabularies and OOV words. We found that vocabularies grow much faster for unconstrained domains such as broadcast news than

it does for smaller specialized domains. We also examined properties of OOV words. We found that OOV words are only slightly longer than IV words, but when the length is weighted by the frequency of usage, OOV words are significantly longer than IV words. We also found that OOV words are mostly nouns, names, and verbs. Our findings were similar to those reported in the literature.

Chapter 4

Modelling OOV Words for Robust Speech Recognition

4.1 Introduction

One of the main challenges in dealing with the OOV problem is to reliably detect and recognize new words without degrading performance on words already in the vocabulary. Previous research has focused on incorporating a simplistic model of OOV words that can easily confuse known words to with OOV words. The approach we present here introduces an *explicit* and *detailed* model of OOV words that has the same level of detail as IV models in order to accurately recognize the underlying structure of the word [Bazzi and Glass 2000b].

This chapter is divided into two parts. In the first part, Sections 2 through 5, we describe the OOV approach and the various model configurations. We start by describing the general framework for our approach. Next, we apply this framework for modelling OOV words within a single stage recognition architecture. We present the FST representation as well as the probability model of our approach. Next, we describe several configurations of the OOV model as well as various techniques to constrain the types of phone sequences that can be recognized as an OOV word [Bazzi and Glass 2001]. In the second part of the chapter, Sections 6 through 9, we describe a series of experiments on the JUPITER and HUB4 domains. We start by describing some performance measures relevant to the OOV problem. We then describe the baseline systems for the two domains. Finally, we present results under a variety of conditions and configurations of the models.

4.2 The General Framework

Embedding knowledge sources such as a vocabulary list or a statistical language model into a speech recognizer is a key element to good recognition performance. The trade-off, however, is that utilizing these knowledge sources constrains the recognizer to the vocabulary and language model and can significantly degrade performance on mismatched cases such as an OOV word or an out-of-domain utterance. At the very heart of this framework is an attempt to retain recognition flexibility while utilizing these knowledge sources.

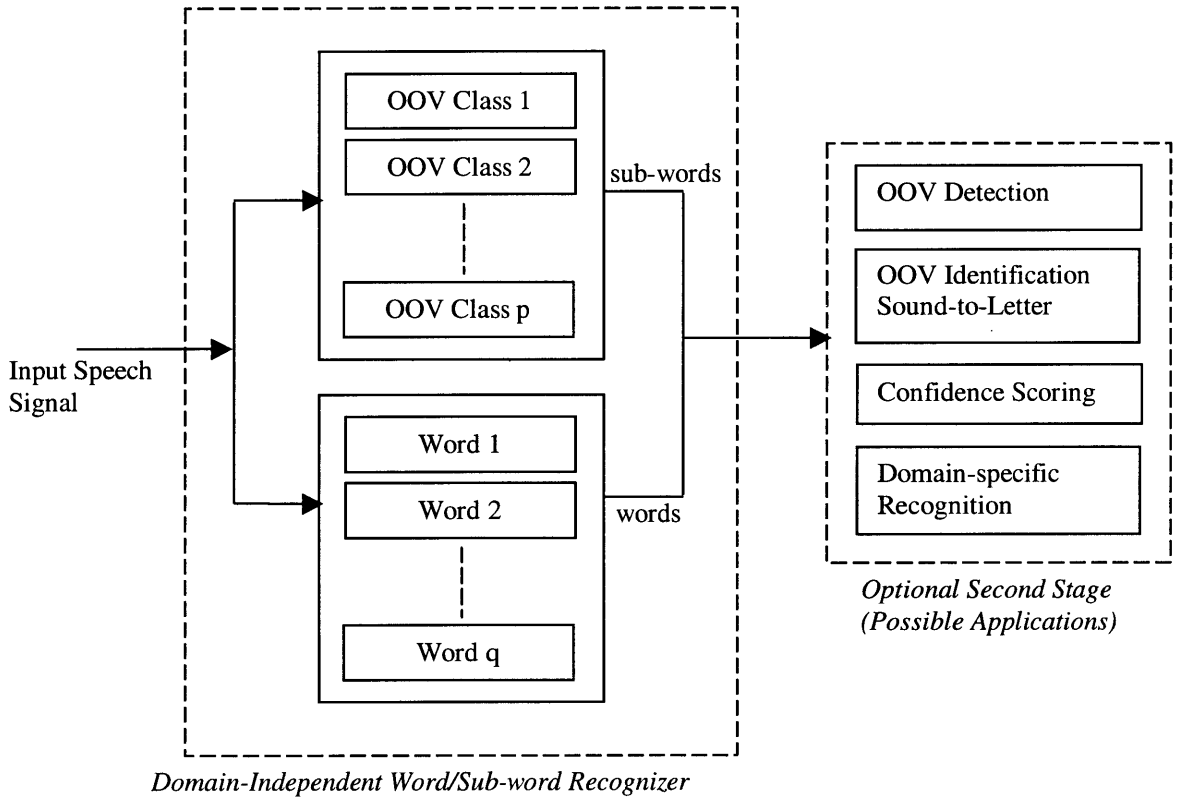

Figure 4-1: The proposed framework.

Figure 4-1 shows the architecture of an open-vocabulary domain-independent recognizer. The two differentiating characteristics of this framework are the lexical component and the language model component.

The lexicon consists of two sets of vocabularies. The first is a list of domain-independent words referred to as $Word_1, ..., Word_q$ in the diagram. The size of this list can range

anywhere from zero to several thousand, covering the whole spectrum from pure subword recognition to large vocabulary speech recognition. The second vocabulary is set of classes or categories of OOV words referred to as *OOV Class* $1, ..., $ *OOV Class* p in the diagram. Examples of such OOV categories are city names, scientific terms, etc. The number of these categories can be as low as one in the case where we are trying to only detect OOV words. The only restriction on the number of classes is the efficiency of the recognition search. Underlying each one of the OOV classes is a subword network that in theory allows for an arbitrary subword sequence. The subwords could be phones, syllables, or any plausible subword unit.

The language model is a combination of two language models. The first is the basic word-level n-gram; the second is a subword language model utilized only when an OOV class is considered during search. The word-level language model includes, in addition to the n-gram probabilities for words in the vocabulary, n-gram probabilities for all supported OOV classes. A key property of this recognition architecture is the hybrid word/subword search space: at the word level, it is guided by the word-level language model (in addition to the acoustic model probabilities) but once in an OOV class model, it is guided by the subword language model to find the best subword realization of the OOV word.

The output from this recognizer can be used for a variety of applications. If we are interested in detecting OOV words, all output hypotheses belonging to OOV classes will be considered OOV detections. If we further want to try and propose an OOV word, the best subword sequence (or a subword graph) can be input to a second stage system for performing sound-to-letter conversion. One interesting application is to integrate this recognizer with confidence scoring to improve the detection of recognition errors. In addition, when domain-specific information is available, a second stage recognizer can be used to improve recognition via N-best rescoring.

4.3 Modelling Out-Of-Vocabulary Words

In this section we apply the framework presented above for the particular task of modelling a single class of OOV words within a word-based recognizer. We describe the three main steps in building such a speech recognizer with explicit knowledge about OOV words.

4.3.1 The IV Search Network

The IV search network is a standard word-based recognizer. The search space allows for any sequence of words from a finite set, those in the vocabulary of the recognition system. Typical recognizer configurations deploy a bigram language model in a forward Viterbi search, and a trigram (or higher-order) language model in a backward A^* search [Glass et al. 1996].

The FST representation is the same as that presented in Section 2.3. We refer to the search space of the word-based recognizer as R_{IV}, given by:

$$R_{IV} = C \circ P \circ L \circ G, \tag{4.1}$$

where C represents the mapping from context-dependent to context-independent phonetic units; P represents the phonological rules; L represents the word lexicon; and G represents the language model. The process of recognition then becomes a search for the best path through R_{IV}. The basic topology of the lexical component of this recognizer, illustrated in Figure 4-2, implies that traversing the network requires going through one or more words in the vocabulary. This is represented with words $w_1, ..., w_n$ for the vocabulary and the loop back transition allowing for an arbitrary number of words.

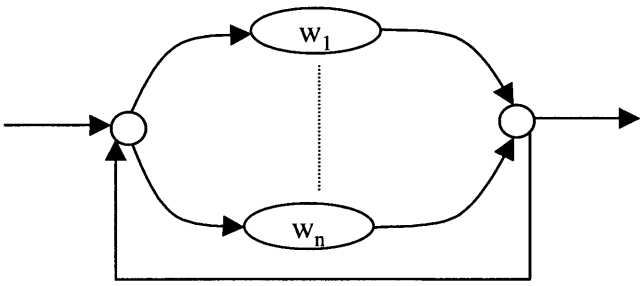

Figure 4-2: The IV search network. Only a finite set of words is allowed. Any sequence of the words can be generated during recognition.

4.3.2 The OOV Search Network

Since an OOV word is by definition a word that we don't know about (i.e, that is not in our vocabulary), a model for such a word must allow for arbitrary phone sequences during recognition. To achieve this goal, an OOV search network should allow for subword units such as phones or syllables. In FST terms, the OOV search network is represented as:

$$R_{OOV} = C \circ P \circ L_u \circ G_u \circ T_u \tag{4.2}$$

where C and P are the same as in R_{IV} indicating that the OOV network uses the same acoustic and phonological models as the baseline or IV recognizer. L_u is the lexicon used to define the subword units used in the OOV network. For example, if the subword units used are the phoneme set, then L_u will be a simple FST mapping each phoneme to itself. However, if the subword units are syllables, then L_u will map each syllable into its phonemic sequence. In other words, this lexicon will represent the pronunciation of syllables in terms of phonemes. Note that in Equation 4.2, composing P with L_u maps the *phoneme* representation in the subword lexicon into the corresponding *phone* representation, hence allowing the OOV model to generate a sequence of *phones* during recognition. Figure 4-3 shows the network corresponding to such a lexical configuration. Similar to Figure 4-2, the search network allows for any sequence of subwords from some inventory $u_1, ..., u_n$.

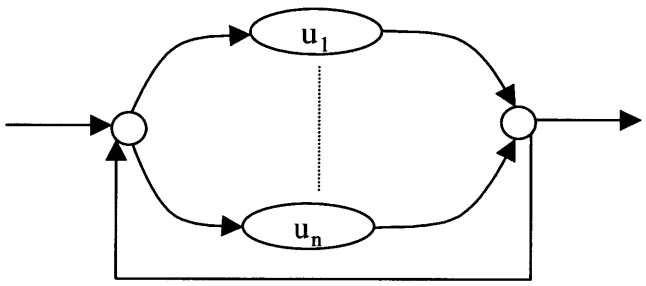

Figure 4-3: An OOV search network that is based on subword units. Any unit sequence is allowed providing for the generation of all possible OOV words.

G_u is the grammar or statistical language model used to constrain the search through the OOV model. Examples include a phoneme-level n-gram language model. G_u biases different paths through the network but does not prohibit any subword sequences. Hence

G_u provides *soft constraints* during the search. T_u, on the other hand is intended to provide *hard constraints* during the search. These constraints include imposing a certain topology on the generated OOV words, such as minimum or maximum length requirements.

4.3.3 The Hybrid Search Network

To create a recognition configuration that supports OOV words, we merge the two search spaces, the IV and OOV networks. The goal is to create two competing branches. The first branch is the IV branch that models the words we know about, i.e., the words in the vocabulary. The second branch is the OOV branch, which can generate phone sequences that are not in our baseline vocabulary. Figure 4-4 shows how the two search spaces are merged. We simply allow the search to transition into the OOV branch W_{OOV}. As we exit W_{OOV}, we are allowed to either end the utterance or enter into any other word, including the OOV word.

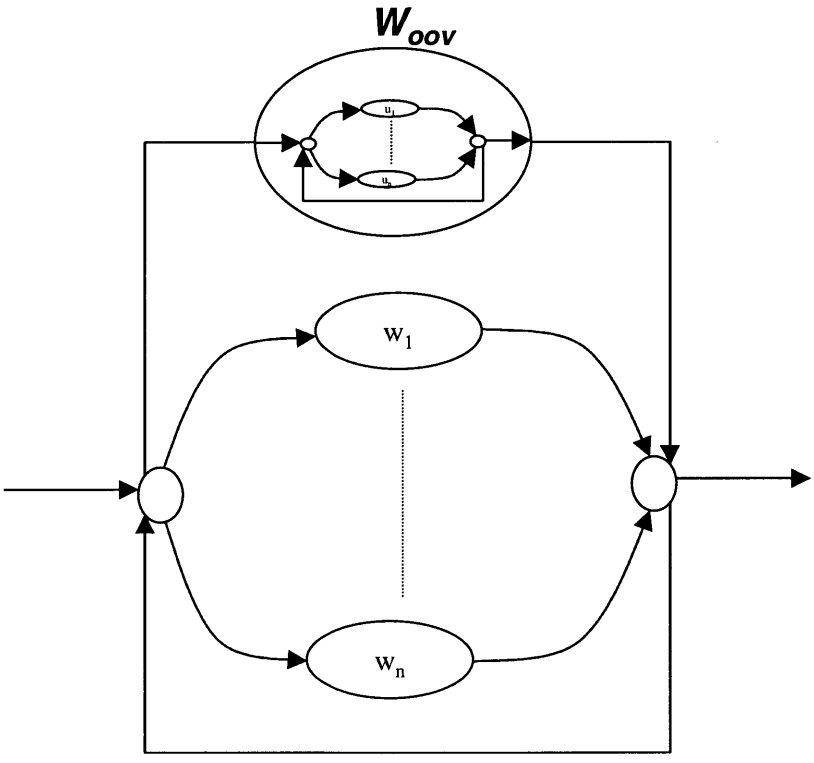

Figure 4-4: The hybrid search network. During recognition, the IV and OOV branches are explored at the same time to allow for OOV recognition

The transition into the OOV model is controlled via an OOV penalty (or cost) C_{OOV}. This penalty is related to the probability of observing an OOV word and is used to balance the contribution of the OOV phone grammar to the overall score of the utterance. For our experimental studies we varied the value of C_{OOV} to quantify the behavior of this approach. The language model of this hybrid recognizer remains word-based, but must now include an entry for OOV words. Since the OOV word is part of the vocabulary, the grammar will include n-grams with OOV words that will be used during the search just like transitions into any other word in the vocabulary.

The hybrid recognizer can be represented with FSTs as follows:

$$R_H = C \circ P \circ (L \cup L_u \circ G_u \circ T_u)^* \circ G \tag{4.3}$$

where R_H is the hybrid search space. G is simply the same as in the IV network except for the extra unknown word in the vocabulary. When an unknown word is encountered in the training of G, the word is mapped to the W_{OOV} tag, which is treated like any other word in the vocabulary. For example, for a bigram language model, G will have bigram pairs such as (W_m, W_{OOV}) and (W_{OOV}, W_n) for some words W_m and W_n in the vocabulary.

The search space R_H is based on the union of the two search spaces. The FST union operation, $\cup$, provides the choice of going through either the word network from the vocabulary or through the generic word network. The $*$ operation is the closure operation on FSTs. This operation allows for switching between the two networks during the same search allowing for the transition into and out of the OOV network as many times as needed.

4.3.4 The Probability Model

The probability model for our approach follows the standard formulation for speech recognition with one key variation. In a typical recognizer, the goal is find the most likely sequence of words $\mathbf{W}$ as described in Section 2.2. This can be written as:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{A}|\mathbf{W})P(\mathbf{W}), \tag{4.4}$$

where $P(\mathbf{A}|\mathbf{W})$ is the acoustic model component that reflects how likely the signal fits a particular sequence of words, and $P(\mathbf{W})$ is the language model component. With the OOV model in place, the goal is still to find the most likely sequence of words but now the

vocabulary we are dealing with is infinite in size. This is because the OOV branch could allow for an infinite number of distinct phone sequences each corresponding to a potential new word. Our formulation is equivalent to treating all possible OOV words as members of one class of words (the OOV class) from a language model point of view, while using the same acoustic models for both IV and OOV words.

To get a better understanding of this formulation, we describe a simple scenario. Consider a part of the acoustic waveform which we will refer to as $\mathbf{A}$. The acoustic score of $\mathbf{A}$ through the IV search network is simply $P(\mathbf{A}|w_i)$ for some word $w_i \in L$ and the language model score is $P(w_i)$ assuming a unigram language model (higher order n-grams follow the same argument). On the other hand, if this waveform is picked by the OOV branch, the acoustic score will be $P(\mathbf{A}|\mathbf{U})$ for some allowable sequence of subword units $\mathbf{U} = u_1 u_2 \ldots u_N$. The language model score will be $P(\mathbf{U}|OOV)P(OOV)$. The first term, $P(\mathbf{U}|OOV)$, is the probability of the subword sequence $\mathbf{U}$ given that we are in the OOV branch. The second term $P(OOV)$ is the probability of entering the OOV branch for a unigram language model. An important observation here is that this formulation for the language model component is the same as that of a class n-gram, where the first term represents the probability of membership in the OOV class and the second term represents the probability of the class. We can see from Equation 4.4 that an OOV word is hypothesized by the recognizer if the following condition is satisfied:

$$\begin{aligned} &\exists \textit{ a subword sequence } \mathbf{U} = u_1 u_2 \ldots u_N \textit{ s.t. } \forall w_i \in L \\ &P(\mathbf{A}|\mathbf{U})P(\mathbf{U}|OOV)P(OOV) > P(\mathbf{A}|w_i)P(w_i) \end{aligned} \tag{4.5}$$

An important point to emphasize is that our approach is *equivalent* to adding an infinite number of words to the vocabulary. By grouping all possible OOV words into the OOV class, we can have a single entry in the recognizer corresponding to possibly an infinite number of words. For higher-order language models the same formulation is used, but now instead of only using $P(OOV)$ we will use $P(OOV|h)$ where h is the word sequence history that depends on the order of the n-gram. Similarly, as we exit the OOV network, the history will include the OOV word.

4.3.5 Comparison with Other Approaches

Our approach is somewhat similar to the filler model approach presented in Section 3.2.4. The idea of having a generic word model that can absorb the OOV portion of the signal is common to the two approaches. However, we can draw on several key properties that differentiate our approach from using filler models.

The acoustic models typically used in filler models are much less detailed than those used in IV words [Asadi and Leung 1993; Jusek et al. 1995]. In most approaches that use filler models, either an all-phone model or a simple context-independent model is used to absorb the OOV part of the signal. In our approach, when the OOV word is considered, the acoustic score is obtained from the same models used for IV words. These models are context-dependent and provide for a more accurate estimate of the acoustic score of the OOV word.

Another important distinction of our approach is the inclusion of OOV words in the word-based language model to predict the presence of such words in relation to other words in the utterance. The work by Jelink *et al.* [Jelinek et al. 1990] was the first to address this issue, but most reported work ignores this aspect of the problem. In our approach, the language model guides the search into and out of the OOV branch using the probability of the OOV word in the context of other words in the utterance.

The use of a statistical language model at the subword level is one of the most important aspects of our approach. This language model provides more constraint during the search that can control what is being generated by the model, resulting in more accurate subword recognition. In addition, the ability to provide hard constraints on the topology of the OOV word ensures that only certain types of phone sequences can be hypothesized.

Augmenting the word recognizer with the generic OOV model is also similar to using filler models for word spotting. In our case, the entire word vocabulary is used in the search, whereas the generic word is intended only to cover OOV words. In most word spotters that use a filler model, the effective vocabulary is much smaller, so that most input words are covered by the filler model. The second distinction is that accurate subword recognition is important for our OOV model since we intend to use its output for a second stage of processing to identify the OOV word. In contrast, word spotters typically make no use of the output of the filler models.

4.4 OOV Model Configurations

There are several requirements for the W_{OOV} model. It must be flexible enough to model the phonetic realization of any word (with the possible exception of the active words in the vocabulary). It must also be accurate, both in its ability to correctly identify the phonetic sequence of an OOV word (possibly for further processing) and in its ability to discriminate between OOV words and IV words. In the following sections we describe three model configurations. First, we describe a baseline corpus model. Second, we describe an *oracle* model which was designed to measure an upper bound on performance. Third, we describe a dictionary model which was designed to be domain-independent.

4.4.1 The Corpus Model

One of the simplest OOV models is a phone recognizer, i.e. a recognizer whose vocabulary is the set of phones in the language. Since the phone unit inventory covers all possible words, we can use it to model the OOV branch of the hybrid configuration. The phone inventory also has the advantage of being small. A model that is based on a phone recognizer is constrained by the phone n-gram language model that biases different paths in the network. The phone-level language model is trained on phone sequences from the same training corpus as the word-level recognizer, but with the words replaced by their phonetic pronunciations. We refer to this type of model as the *corpus model* because its subword language model is trained on a corpus of running text. The phonotactic constraints through the OOV branch are therefore based on the statistics of a training corpus.

As for the FST representation, L_u represents the phoneme set of the recognizer, G_u is a phoneme n-gram trained from some corpus, and T_u is used here to ensure a minimum number of phonemes. Applying the phonological rules to the model maps the representation from the phoneme to the phone level.

4.4.2 The Dictionary Model

The motivations for this model are inspired by several drawbacks of the corpus model. First, since the corpus model is trained on phonetic transcriptions of sentences in the training corpus, the n-gram probabilities are influenced by the frequency of words in the corpus and will obviously favor more frequent words (e.g., *the*, *is*, and *at*). Second, in addition

to modelling word-internal phonetic sequences, the n-gram would devote probability mass to cross-word sequences. Clearly, neither of these properties is desirable for modelling rare OOV words. A third issue with the corpus OOV model is the domain-dependent nature of the training corpus. Since OOV words are often out-of-domain, a more domain-independent approach is desirable for training the subword n-gram.

The dictionary model is different from the corpus model in the way we train the n-gram language model, G_u. For the dictionary model, the OOV phoneme n-gram is trained from a dictionary instead of a corpus of utterances. In this dictionary-based approach, we estimate the n-gram from phoneme sequences in a large domain-independent word dictionary. This dictionary is significantly larger than the word vocabulary of the recognizer. By using a large vocabulary, we reduce domain-dependence bias; by training on vocabulary items, we avoid modelling cross-word phonotactics, and eliminate biasing the OOV network toward frequent words (i.e., atypical OOV words). L_u and T_u are the same as those of the corpus model.

4.4.3 The Oracle OOV Model

There is one interesting configuration of the OOV model that can give some insights into the best possible performance that can be obtained with this hybrid approach. We define the *oracle* model as a model that knows which OOV words will be encountered during recognition and allows only the phone sequences corresponding to to these words.

To build such an oracle model, all OOV words in the test set are identified and their phone sequences are used to build the OOV search network. Such a network is unfair because it knows about the OOV words that will be encountered during recognition. Its only purpose is to provide an approximate upper bound on performance. The oracle OOV configuration is different from simply adding the OOV words to the recognizer vocabulary for two reasons. First, the n-gram probabilities will be those of the general OOV word as opposed to the n-gram probabilities of each word. Second, the cost of entering the OOV network C_{OOV} controls how often an OOV word is selected, which also changes the behavior of the recognizer.

As for the FST representation, L_u consists of the phone sequences of all OOV words in the test set. G_u and T_u are not used with this configuration.

4.5 Applying Topology Constraints

Imposing constraints on the topology of the OOV model can be useful for many reasons. It allows us to explicitly incorporate certain facts about OOV words, for examples that an OOV word should be at least 3 phones long. In this section we look at possible topology constraints: what T_u should be and how to build it. The main goal is to provide the *hard constraints* that can prohibit certain phone sequences from being generated. We focus on two types of constraints: length constraints, and *novelty* constraints that have to do with prohibiting IV words within the OOV search space.

4.5.1 OOV Length Constraints

A key characteristic of OOV words is that they tend to be *long* content words. A model that is simply based on a phone loop could allow for words as short as one or two phones, very unlikely OOV words. To prohibit such behavior, we construct T_u such that $P(\mathbf{U}|OOV) = 0$ if the length of the subword sequence $\mathbf{U}$ is less than n phones. The FST T_u is shown in Figure 4-5 for $n = 3$. In this representation, the notation *a:b* is used where a is the input symbol and b is the output symbol of the transition. In this figure, all transitions have the symbol p representing any phone in the phone set. Note that the output side should have the same symbol as the input side because such an FST is only intended to provide length constraints without altering the phone sequence. Also note that if a phone sequence is less than 3 phones long it cannot get to the final (or accept) state of the FST, and hence cannot be generated during recognition.

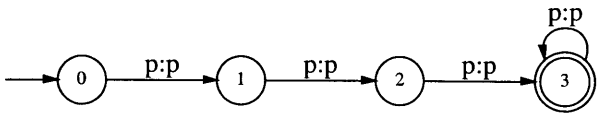

Figure 4-5: An FST T_u that enforces a minimum of $n = 3$ phones for an OOV word. All words with less than 3 phones are prohibited.

Similarly, we can constrain the maximum length of an OOV word. Figure 4-6 shows an FST T_u where words are restricted to be between 3 and 5 phones in length. Note that from states 3 and 4, a transition can happen into the final state without any input and generating no output. This is represented with the ϵ symbol on both the input and the

output side of the transitions. Second, there is no self-loop on the final state indicating that a phone sequence longer than 5 phones will not be allowed through the network. The main reasoning behind such a topology is to prohibit the OOV model from mistakenly absorbing a large portion of the waveform.

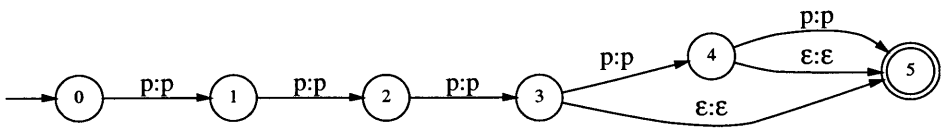

Figure 4-6: An FST T_u that allows for words between 3 and 5 phones long. All words less than 3 phones or more than 5 phones in length are prohibited.

4.5.2 The Complement OOV Model

An attractive feature of an OOV model is the ability to generate only *novel* phone sequences. A novel phone sequence is one that does not match any phone sequence in the IV branch. Generating only novel phone sequences guarantees that the hypothesized OOV word is a new word with a new phone sequence that does not belong to the vocabulary.

It is known from finite-state theory [Sipser 1997] that for every finite-state machine that recognizes some regular language L, there exists a finite-state machine that can recognize the complement language $\bar{L}$. The construction of such a machine involves swapping final and non-final states, in addition to some other details depending on the type of the finite-state machine. This property can be used to construct a *complement* OOV model. If L represents the lexicon of the IV branch, then we can let the topology constraints $T_u = \bar{L}$, hence the hybrid search space becomes:

$$R_H = C \circ P \circ (L \cup L_u \circ G_u \circ \bar{L})^* \circ G \tag{4.6}$$

To get some intuition into how $\bar{L}$ is constructed, we describe a simple example. Assume that the phone set, referred to as Σ, contains only the two phones denoted by a and b. Also assume that the lexicon consists of only two words, whose pronunciations are aa and ab. This is represented formally as:

$$\Sigma = \{a, b\} \tag{4.7}$$

$$L = \{aa, ab\} \tag{4.8}$$

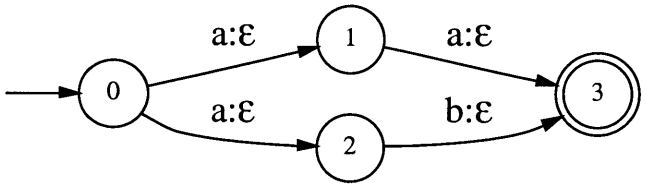

Figure 4-7: Sample lexicon FST L. Only the two words $w_1 = aa$ and $w_2 = ab$ are in the vocabulary.

The FST for this lexicon is shown in Figure 4-7. As can be seen from the FST, only the two words *aa* and *ab* are allowed. The complement FST is shown in Figure 4-8. In order to construct the complement FST, all final and non-final states are swapped. In addition, a *dead* state is added, and transitions for un-accounted for symbols (phones) are created from all states into this dead state. Note that if we try to walk the FST starting at the initial state with input *aa* or *ab*, we will end up in a non-final state; hence the two sequences are prohibited. However, if we try to walk the FST with sequences such as *a*, *aab*, or *bbb*, we will end up in a final state. Formally this complement FST is represented as:

$$\bar{L} = \Sigma^* - \{aa, ab\} \tag{4.9}$$

This example illustrates how we can take a lexicon and create from it an OOV search network that prohibits IV phone sequences. An important point to note here is that the size of the complement model is on the same order as that of the lexicon, while for the corpus and dictionary models, the model size is significantly smaller than that of the IV search network since it consists of a phone-level lexicon and language model.

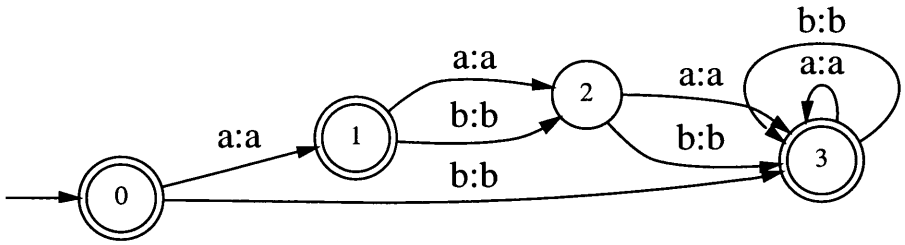

Figure 4-8: The complement lexicon FST $\bar{L}$. This FST allows for all phone sequences except *aa* and *ab*.

4.6 Performance Measures

In evaluating approaches for handling OOV words, we focus on four different performance measures. Each one of the measures is important depending on the type of application the approach is used for. In the following four sections, we describe each measure and comment on its importance and usage.

4.6.1 Detection and False Alarm Rates

For applications where we are simply interested in knowing whether an utterance or a word is not in our vocabulary, an important measure of performance is the OOV detection quality. In detection problems, the quality of detection is expressed in terms of two parameters, the *detection rate* (DR) and the *false alarm rate* (FAR). The detection rate is the ratio of the number of correctly detected OOV words to the total number of OOV words. The false alarm rate is the ratio of the number of wrongly hypothesized OOV words to the total number of IV words:

$$DR = \frac{Count(OOV_{cor})}{Count(OOV_{ref})} \tag{4.10}$$

$$FAR = \frac{Count(OOV_{incor})}{Count(IV_{ref})} \tag{4.11}$$

Since there are many more IV words than OOV words, a high FAR indicates that a large number of IV words are mistakenly recognized as OOV words. Because our goal is to detect as many OOV words as possible without degrading performance on IV words, we are

interested in very low false alarm rates. As we vary the cost of entering the OOV model, both DR and FAR vary. The relationship between the two measures is referred to as the *receiver operating characteristic* or ROC. The ROC curve is a plot of the DR versus FAR over the range of 0% to 100%. In most of our results, we will show the ROC for regions of low FAR (less than 10%), the region of most interest to us.

A single measure that combines DA and FAR across all operating points on the ROC curve is the *figure of merit* (FOM). FOM is the area under the ROC curve. The higher the FOM the better the performance of the system. For example, an FOM of 1 means perfect detection with no false alarm.

DR and FAR are not the only measures used to evaluate the detection quality. A closely related set of measures are *recall* and *precision*. Recall is the same as DR above, while precision is the ratio of correctly detected OOV words to the total number of hypothesized OOV words. In all of our reported results, we use the DR and FAR described above.

4.6.2 Recognition Accuracy

In addition to the quality of the detection of new words, we examine the impact of the OOV model on the word error rate (WER). An important goal of the approach is that it should not degrade performance on utterances with no OOV words, because those make up the majority of the utterances encountered by the system. We report word error rates on the complete test sets as well as on the IV subset of the test set.

4.6.3 Location Accuracy

Another important performance measure is the accuracy of the system in locating the OOV word. Given the true start and end times of a word as well as the start and end of the hypothesized OOV word, a common measure is the time shift s between the true and hypothesized boundaries. This is illustrated in Figure 4-9.

Location accuracy is typically expressed with the cumulative statistics of s. This is the fraction of detected words where the shift s falls within some window T. This can also be expressed as $P(s < T)$. Typical values of T are between $0msec$ and $100msec$. For example for $T = 50msec$, $P(s < T = 50\ msec)$ is the fraction of detected words within 50 msec of the true boundary. This quantity can also be conditioned on whether the boundary is a start or end of a word. We will report results under both conditions.

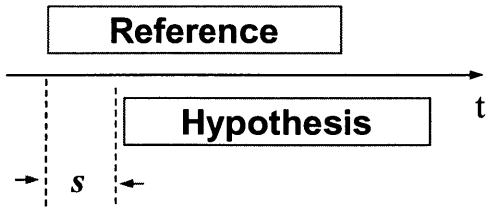

Figure 4-9: The shift s is the difference between the true start (or end) of a word and the hypothesized start (or end) of the word.

4.6.4 Phonetic Accuracy

A fourth performance measure is the phonetic accuracy of generated new words. The standard measure is the phonetic error rate (PER), defined as the sum of substitutions, deletions, and insertions of phones in the most likely pronunciation of the true word when compared to phonetic sequence generated by the recognizer. This PER measure is particularly important for applications that try to identify the OOV word in a second stage of recognition. The measure is also important for applications that try to look up the word in some larger dictionary or to generate a spelling of the word using a sound-to-letter approach [Meng et al. 1994; Meng 1995].

4.7 Experimental Setup

In this section we describe the baseline systems of the two domains, including performance with a baseline closed-vocabulary system that does not support OOV words.

4.7.1 JUPITER Baseline

The acoustic models for the baseline recognizer were trained on the the full training corpus of 88,755 utterances. Refer to Section 2.4.1 for a complete description of the corpus. The test set contained a total of 2,029 utterances (11,657 words), 314 of which contain OOV words. Most of the 314 utterances had only one OOV word. At the word level, the OOV

rate was 2.2%.

The features used in the acoustic models are the first 14 Mel-frequency cepstral coefficients (MFCC's) averaged over 8 different regions near each boundary in the phonetic segmentation of each utterance [Glass et al. 1999]. These 112 features (14 $MFCC's \times$ 8 *regions*) are then reduced to a 50-dimensional vector using principal components analysis (PCA). The PCA transforms the resulting feature space, producing an identity covariance matrix for the training data [Bishop 1995]. The acoustic models are diagonal Gaussian mixture models with up to 50 mixture components per model, depending on the amount of training data available. The mixture Gaussian models are initialized using K-means clustering [Rabiner and Juang 1993] followed by estimation of each Gaussian's parameters via the expectation-maximization (EM) algorithm [Dempster et al. 1977]. The boundary labels for the acoustic models are based on a set of 62 phone labels. The total number of acoustic models used was 1,092, including both internal and transition boundaries. The number of models is determined by a decision tree clustering procedure of all boundary models encountered in the vocabulary. Boundaries with few or no training tokens are grouped with similar boundaries to form boundary classes with enough training data.

The word-level lexicon consisted of 2,009 words, many of which have multiple pronunciations. The baseline recognizer uses a word bigram for the first pass Viterbi search and a word trigram for the second pass A^* search. For training the n-grams, we used the 88,755 utterances, which include both in-vocabulary and out-of-vocabulary utterances. All out-of-vocabulary words were treated as a single word during training and mapped to the token W_{OOV}.

Set	Substitutions	Insertions	Deletions	WER
complete test	7.3	3.6	6.2	17.1
IV only test	5.9	2.4	2.6	10.9

Table 4-1: Rates of substitution, insertion, and deletion and word error rates (WER) obtained with the JUPITER baseline recognizer on the complete test set (2,029 utterances) and on the IV portion of the test set (1,715 utterances).

Table 4-1 shows the error rates obtained using the baseline recognizer on the complete test set and on the IV portion of the test set. In all cases, the beam width in the Viterbi search, measured in terms of the score difference between the best- and worst-scoring path

considered, is 25.

The OOV model used a lexicon of 62 entries, the complete phone set of the system. For configurations that use a phone n-gram, we trained a bigram language model, either from the complete training corpus (for the corpus model) or from a large lexicon for the dictionary model. For all our experiments where a large lexicon is used, such as in the dictionary OOV model, we used the LDC PRONLEX [McLemore 1997] dictionary, which contains 90,694 words with 99,202 unique pronunciations.

4.7.2 HUB4 Baseline

The acoustic models for the HUB4 baseline recognizer were trained on the 1996 and 1997 acoustic training corpus [Graff and Liberman 1997], using all 106 hours of recorded speech. The training process is the same as the one described for the JUPITER baseline in the previous section. There were 1,573 boundary models. The lexicon was derived from the acoustic training data. We built a recognizer that had complete coverage of words in the training set. The number of words in this lexicon was 24,771. We trained the language model on 160 million words, a significantly larger amount of data than that used in the acoustic training. The data was collected from various sources including the Wall Street Journal and various transcripts from television and radio shows on stations such as CNN and NPR [Kubala et al. 1997]. The baseline recognizer uses a word bigram for the first pass Viterbi search and a word trigram for the second pass A^* search.

Set	Substitutions	Insertions	Deletions	WER
F0 condition	15.8	5.0	4.1	24.9

Table 4-2: Rates of substitution, insertion, and deletion and word error rates (WER) obtained with the HUB4 baseline recognizer on the F0 condition.

We ran experiments only on the F0 condition, prepared and clean speech. The test set consisted of a total of 10,164 words broken into 192 *long* utterances. Because the utterances were long and had a high OOV rate (2.7%), we did not break the test set into IV and OOV subsets as we did in the JUPITER case. Had we done that, most of the utterances would be in the OOV subset. Table 4-1 shows the error rates obtained using the baseline recognizer on the F0 condition. In all cases, the beam width in the Viterbi search, measured in terms of

the score difference between the best- and worst-scoring path considered, was 12. Because of the larger vocabulary and language model, the beam width was set much smaller than for the JUPITER baseline.

The baseline performance of 24.9% is quite high when compared to state-of-the-art performance on the F0 condition [Woodland et al. 1997]. The two main reasons behind this performance are the acoustic models and the lexicon. Our acoustic models are much less detailed than those reported in the literature. Only diphone models were used and the number of models was very small. The lexicon we used had a little less than 25,000 words. Most of the reported research used 64,000 or more words for the lexicon. These experiments were our first attempt at dealing with large vocabulary recognition tasks within the SUMMIT recognition system, hence our choice of the number and type of acoustic models and the size of the lexicon. Our main goal in reporting this work here is to explore the effectiveness of the OOV approach in large vocabulary environments.

4.8 JUPITER Results

The following sections describe in detail the results on the JUPITER domain. The first four sections present the four performance measures described in Section 4.6. The last section describes the impact of applying topology constraints on the model.

4.8.1 Detection Results

The detection quality of the various OOV models was measured by observing the OOV detection and false alarm rates on the test set as C_{OOV} was varied. The presence or absence of an OOV word was based on the orthography of the top recognizer hypothesis. If a hypothesized OOV word is aligned with an unknown word, this is considered a correct detection, while if it aligns with a known word, it is considered a false alarm. Figure 4-10 plots the ROC curves for the three models: corpus, dictionary, and oracle. Note that the origin of the curve (DR=FAR=0) corresponds to $C_{OOV} = -\infty$, a closed-vocabulary system where OOV words are not allowed. The ROC curve is generated by varying C_{OOV} from $-\infty$ to $+\infty$.

Figure 4-10 shows that for the corpus model, the system can detect about half of the OOV words with a FAR of 2%. For a detection rate of 70%, the FAR goes up to 8.5%. The

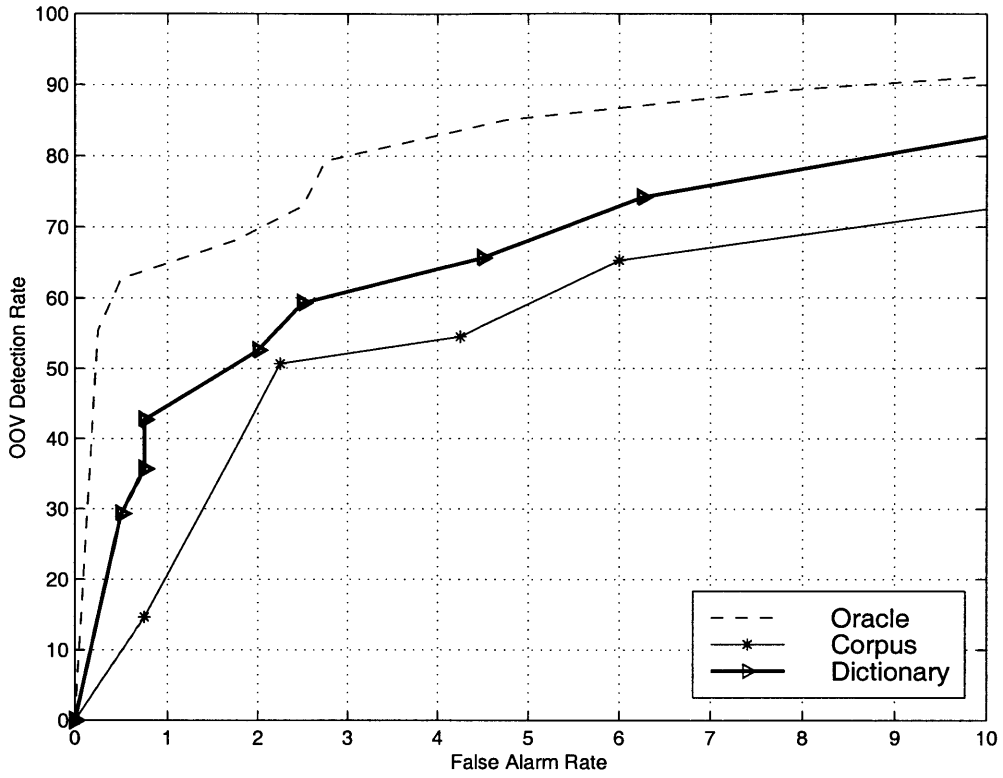

Figure 4-10: ROC curves for the three models: corpus, dictionary, and oracle. Each curve shows the DR versus the FAR for each of the models.

oracle model has significantly better performance. It can detect half of the OOV words with less than 0.25% FAR, and achieves a detection rate of 70% with a FAR of 2%. In addition, the figure shows that the dictionary model improves detection significantly over the corpus system. Half of the OOV words are detected with a 1.5% FAR and a detection rate of 70% is achieved with a 5.3% FAR.

In order to quantify the ROC behavior, we computed the FOM for the three models. For our work we are most interested in the ROC region with low false alarm rates, since this produces a small degradation in recognition performance on IV data. For this reason we measured the FOM over the 0% to 10% false alarm rates. Note that the area is normalized by the total area in this range to produce an FOM whose optimal value is 1. For reference, a randomly guessing OOV model would produce an ROC curve which is a diagonal line (i.e., $y = x$). The FOM over the entire false alarm range would be 0.5, and the FOM over the 0% to 10% false alarm range would be 0.1. Table 4-3 summarizes the FOM measure for the various conditions, both for the 0 to 10% range and the entire ROC area. All of our

following discussion refers to the second set of FOM numbers (the first 10% of the ROC curve).

OOV Model	100% FOM	10% FOM
Corpus	0.89	0.54
Dictionary	0.93	0.64
Oracle	0.97	0.80
Random	0.50	0.10

Table 4-3: The figure of merit performance of various OOV models. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM).

As expected, the oracle OOV network performs best in detecting OOV words, achieving an FOM of 0.80. This FOM gives an approximate upper bound on performance and gives some insight into how much we can possibly improve on our baseline FOM of 0.54. The oracle OOV model we used was clearly sub-optimal; better performance would have been achieved if sentence-specific oracle OOV models were used containing only sentence-specific OOV word(s). The joint oracle OOV model was used since it was easier to compute, and provided at least a lower bound on optimal performance.

With the dictionary model, the FOM improves from 0.54 for the baseline to 0.64. This is an improvement of 19% in the detection quality of the system. This result is quite encouraging because of the domain-independent nature of the vocabulary used to train the dictionary model n-gram. Unfortunately, we cannot quantify the individual contributions of 1) moving from continuous to isolated word n-gram training, and 2) moving from domain-dependent to domain-independent training, since the dictionary model differs from the corpus model in both of these factors. The latter factor would be difficult to quantify with the data we have at our disposal, since we do not have a large number of OOV words in our training data.

4.8.2 Impact on Recognition Performance

Figures 4-11 and 4-12 show the impact of introducing the OOV model into the closed vocabulary system. The first figure shows the impact on the IV subset of the test set and the second figure shows the impact on the overall test set. The figures show results for the

dictionary OOV model.

For all OOV models, the relationship between overall OOV false alarm rate and word error rate (WER) on IV test data is approximately linear. In the case of the dictionary-based OOV model, for example, the WER increases slowly from the baseline WER of 10.9% at 0% false alarm rate to under 11.6% at 10% false alarm rate.

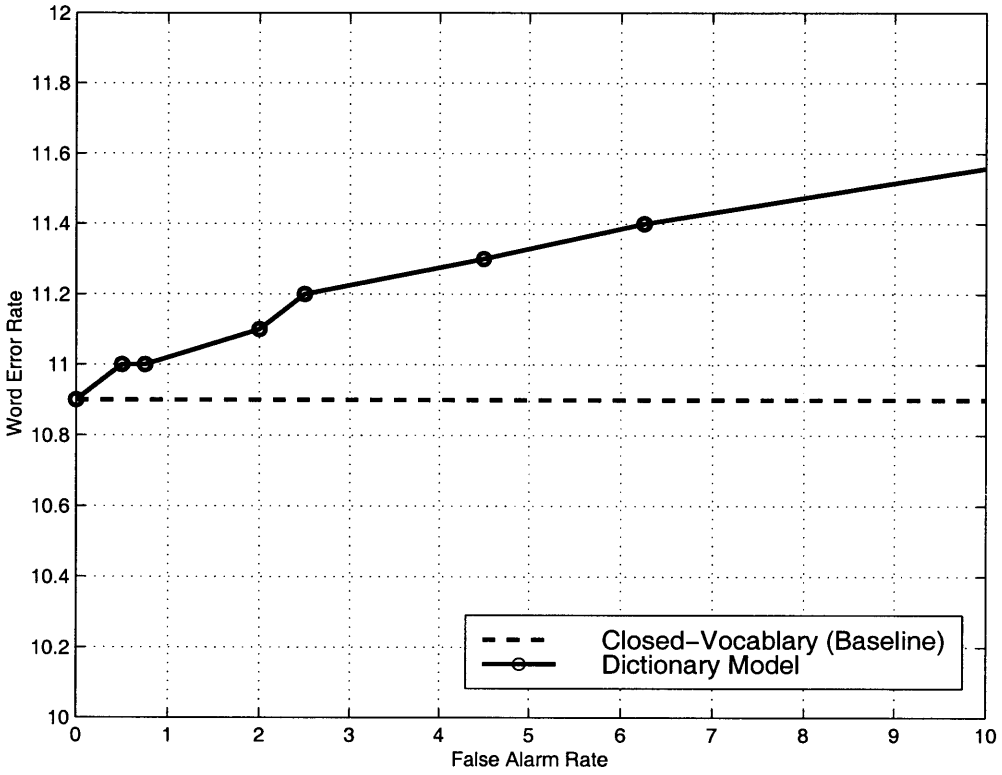

Figure 4-11: WER on the IV test set. Shown is the closed-vocabulary performance (10.9%) and the performance of the dictionary model system as a function of the FAR.

As for the overall WER on the complete test set, starting at the closed-vocabulary WER of 17.1%, the WER decreases as we favor the OOV branch more (with increasing FAR) as seen in Figure 4-12, and then starts to increase again after it hits a minimum of 16.4%. This better performance is due to correcting errors within OOV utterances other than the OOV word itself. In computing the WER, we *do not* assume that a correct detection is a correct recognition, so the improvement in WER is due to correcting words neighboring OOV words. If correctly detected OOV words are counted as correctly recognized, the WER will be 15.1% instead of 16.4%, this is a total of 2% absolute reduction in WER from

the closed-vocabulary WER of 17.1%. As the FAR increases, the overall WER increases as more and more IV words are confused with OOV words. At 10% FAR, the WER jumps to 18.7%.

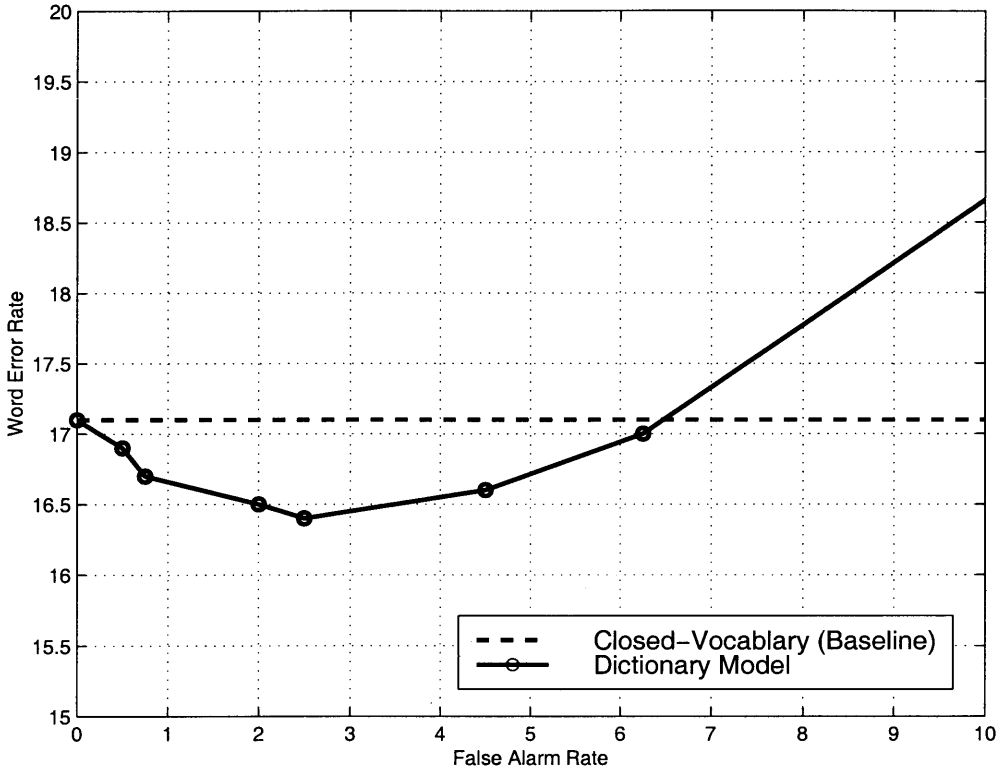

Figure 4-12: WER on the complete test set. Shown is the closed-vocabulary performance (17.1%) and the performance of the dictionary model system as a function of the FAR.

4.8.3 Locating OOV Words

Here we look in more detail at the output of the OOV system. In particular, we report on the accuracy of our approach in finding the start and end times of correctly detected OOV words. This study is conducted on the dictionary OOV model with a cost $C_{OOV} = 0$.

When examining timing information, we define a *tolerance* window (or shift) s on what we consider correctly located. For example, if we assume a tolerance of 25 msec then any detected OOV word is considered correctly *located* if its start and end times are within 25 msec of the true start and end of the word. Our experimental study examines location performance as a function of this tolerance window s.

When an OOV is hypothesized, we can identify three types of behaviors of the system:

Case 1: Either the start time or the end time of the detected word aligns with the start time or end time of the true word within the tolerance s. In this case, the detected word may align with 1 word, 2 words, or any number of words including a fraction of a word which happens when only the start or the end aligns with a true boundary but not both. This case is illustrated in the first plot (highest solid curve) of Figure 4-13. The plot shows, as a function of s, the percentage of boundaries within the window shift of s.

Case 2: Both the start and the end times of the detected word align with the true start and end time of some words in the reference within the tolerance s. In this case, the detected word may align with zero word (when both start and end align with the same boundary) or one or more words but not a fraction of a word. This case is illustrated in the second plot of Figure 4-13. This condition also includes cases where the OOV model absorbs two or more words including the OOV word.

Case 3: This is the subset of case 2 that includes only those detected words whose start and end times align with the start and end of the OOV word in the reference. This case is illustrated in the third plot of Figure 4-13. This is the case that corresponds to an accurate location of an OOV word.

For example, Figure 4-13 shows that if the window shift tolerance s is $50msec$, then 93% of the detected boundaries are within 50 msec of the correct boundary. However, 82% of detected words have both their start and end times within 50 msec of the true times. Finally, 61% of detected words align with the true OOV word at both the start and end times for this tolerance window shift of 50 msec. These results show that the majority of the OOV words are actually detected within the right region in the utterance. The fourth plot in Figure 4-13 shows the fraction of correctly aligned words that do correspond to an OOV word. As we note from the plot, around 75% of those words are correctly aligned with the OOV word for different values of the tolerance shift s.

Next we examine **Case 3** in more detail as this is the correct location case. Figures 4-14 and 4-15 show the histograms and the cumulative distribution functions(CDFs) of starts and ends of correctly detected OOV words as a function of the tolerance s. From Figure 4-14 we can see that most detected words have their boundaries within a few milliseconds of the true boundary. Figure 4-14 shows the fraction of word start and end as a function of the tolerance shift. We can make two main observations from this figure. First, most of the

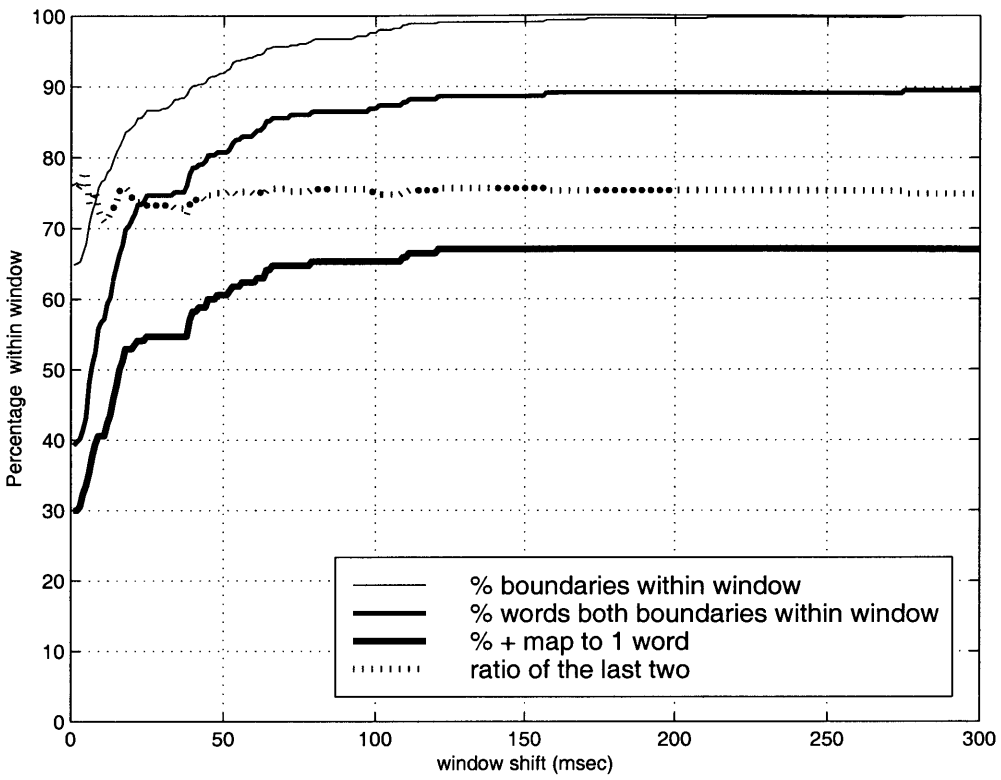

Figure 4-13: A breakdown of the system's performance in locating OOV words. The first plot shows, as function of the shift (or tolerance) s, the fraction of words with start *or* end aligning with a true boundary. The second plot shows the fraction of words where *both* start and end align with true boundaries. The third plot shows the fraction of words where both start and end align with true boundaries *and* of the OOV word. The last plot (dotted) shows the ratio of the third to the second plot.

boundaries (over 80%) are within 20 msec of the true boundary. Second, the word end is located slightly better than the word start, as seen from the CDF plot. One hypothesis is that this difference in behavior may be attributed to the way FST weights are optimized and pushed to the beginning of the search space, resulting in different results for starts and ends of words.

4.8.4 Phonetic Accuracy

In order to measure the phonetic accuracy, we examine the top phone-level hypothesis generated by the OOV model. The results we report here also use the dictionary model with the same conditions as the previous section. Performance, in terms of phonetic recognition error rate, is measured on a collapsed set of 39 classes typically used in reporting phonetic

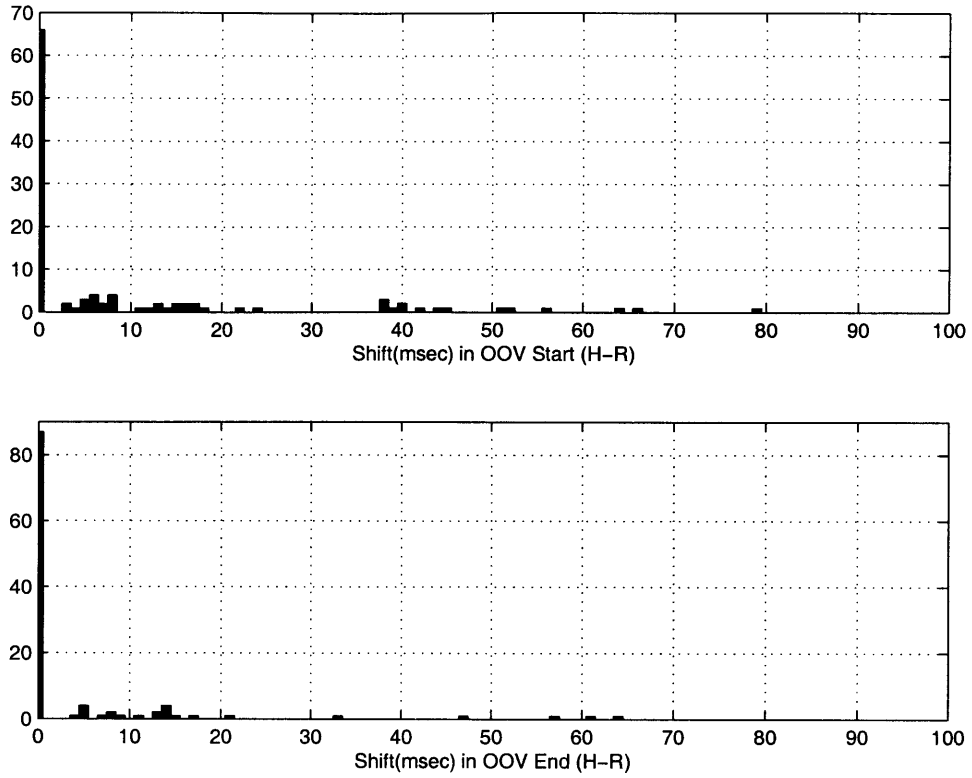

Figure 4-14: Histograms for start (top) and end (bottom) shifts of correctly detected OOV words.

recognition results [Chang and Glass 1997; Halberstadt 1998; Spina and Zue 1997]. Table 4-4 shows the phonetic error rates (PER) obtained for three different shift tolerance windows: 25, 50, and 100 msec.

Shift	Substitutions	Insertions	Deletions	PER
25 msec	18.3	12.9	6.0	37.9
50 msec	18.4	14.1	5.8	38.3
100 msec	17.8	18.0	5.4	41.2

Table 4-4: Rates of substitution, insertion, and deletion and phonetic error rates (PER) obtained with the dictionary model for the three shift tolerance windows of 25, 50, and 100 msec.

Note that the higher the shift, the more words are included in our PER results, so the numbers shown are not for the same set of OOV words. Rather, the first row (25 msec) represents a subset of the second row (50 msec) and so on. In addition, the PER increases

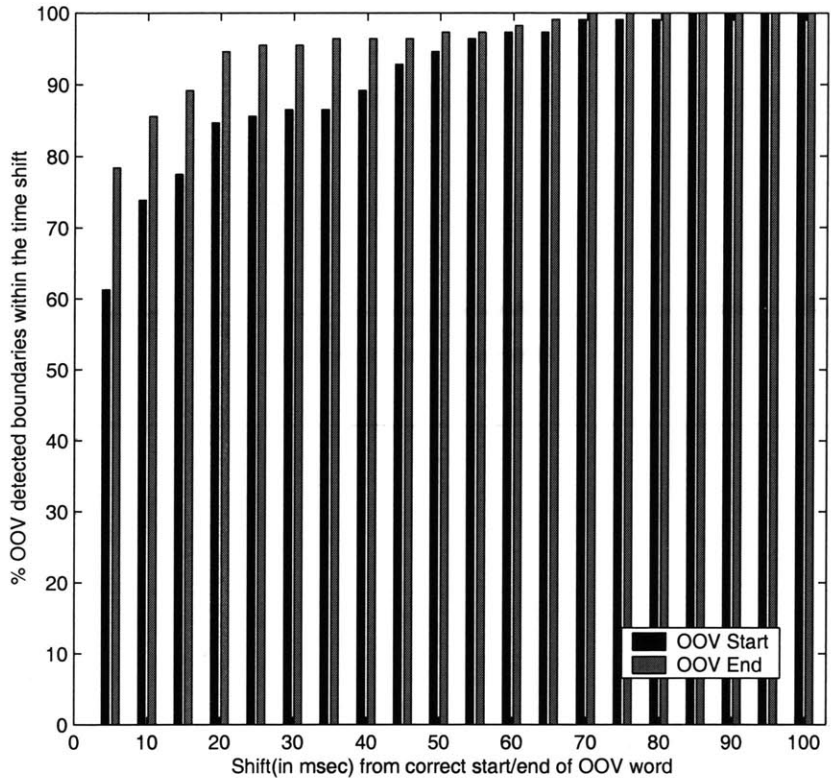

Figure 4-15: Cumulative distribution function of the shift for starts and ends of correctly detected OOV words.

as we include more words with larger shift tolerance. This is predictable since the higher the allowed shift, the less accurate the location of the OOV word, allowing for deleting phones or absorbing phones from neighboring words. The PER of 37.8% for such a task is quite encouraging considering two facts. The first is that we are using only a phone loop with a phone bigram for the OOV model. Second, phonetic recognition tends to be much less accurate than word recognition, and in this case it is particularly hard because of the nature of OOV words and the errors introduced at the boundaries.

For applications that can make use of the phonetic sequence generated by the model, several enhancements can be made to reduce the PER, including recognition from a phonetic graph instead of only the top phone hypothesis. In addition, using higher order phone n-grams may improve phonetic recognition performance.

4.8.5 Imposing Topology Constraints

In this section we report results from experiments on using topology constraints on the OOV model. First we discuss results on applying length constraints on the dictionary model. This included imposing a minimum duration of n phones on the length of the OOV word. We experimented with values of n between 2 and 5. For all conditions, we found that the minimum duration constraint did not change the overall performance of the system, although in very few cases, it removed false alarms where short function words were hypothesized by the model with no minimum duration constraint. Table 5-6 summarizes the results for n between 1 (baseline) and 5. Note that as n increases, performance gets worse than the baseline as some OOV words are actually shorter than 5 phones and cannot be detected with such a constraint.

Min num phones	100% FOM	10% FOM
1 (baseline)	0.93	0.64
2	0.93	0.64
3	0.94	0.64
4	0.93	0.63
5	0.90	0.62

Table 4-5: The figure of merit performance of various minimum length constraints. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM). The baseline is the dictionary model.

Similar behavior is observed when imposing maximum length constraints. We experimented with maximum lengths of 12 and 15 phones per word. This constraint helped eliminate the few cases where the OOV model absorbed several words, or even the whole utterance. However, the ROC curves and the FOM measures were very close to those of the model that does not have this constraint.

Finally, the complement OOV model also had very little impact on the overall performance of the system. Results are summarized in Table 4-6. The main conclusion we can draw from this result is that the model is not hypothesizing within vocabulary phone sequences, and that the phone sequences generated are mostly novel sequences corresponding to new words. Due to the high computational cost of such a model, we did not use it for the rest of our experiments since performance gains were very small.

OOV Model	100% FOM	10% FOM
Dictionary	0.93	0.64
+ Complement	0.94	0.64

Table 4-6: The figure of merit performance of the complement OOV model. This model is obtained by creating a search network that prohibits IV words.

4.9 HUB4 Results

Our main goal for experimenting with the HUB4 domain is to find out if our approach generalizes well to large vocabulary environments. The HUB4 baseline recognition system contains more than 10 times the number of words in the JUPITER recognizer (24,771 words for HUB4 versus 2,009 words for JUPITER). In addition, the HUB4 domain is highly unconstrained and OOV words can appear anywhere in the utterance. In JUPITER, many of the OOV words are city names and tend to come later on in a question about the weather.

Figure 4-16 shows the detection results for HUB4 together with those for JUPITER, both using the dictionary model. The performance on HUB4 is similar to that on JUPITER, although not as good. However, it demonstrates that the approach works well for large vocabulary unconstrained tasks. We attribute the worse performance on HUB4 to the fact that the WER is higher on HUB4 (24.9%) than JUPITER (17.1%). Having more recognition errors means that more words could be hypothesized as OOV words.

Figure 4-17 shows the WER on the complete test set. The results are similar to those we reported for JUPITER. Starting at the closed-vocabulary WER of 24.9%, the WER decreases as we favor the OOV branch more (with increasing FAR). The WER then starts to increase again after it hits a minimum of 23.5%. Similarly to JUPITER, this better performance is due to correcting words within OOV utterances other than the OOV word itself. Also similarly to the JUPITER results, in computing the WER, we *do not* assume that a correct detection is a correct recognition, so the improvement in WER is due to correcting words neighboring OOV words. If correctly detected OOV words are counted as correctly recognized, the measured WER will be 22.5% instead of 23.5%, this is a total of 2.4% absolute reduction in WER from the closed-vocabulary WER of 24.9%. As the FAR increases, the overall WER increases as more and more IV words are confused for OOV words. At 10% FAR, the WER jumps to 33.7%.

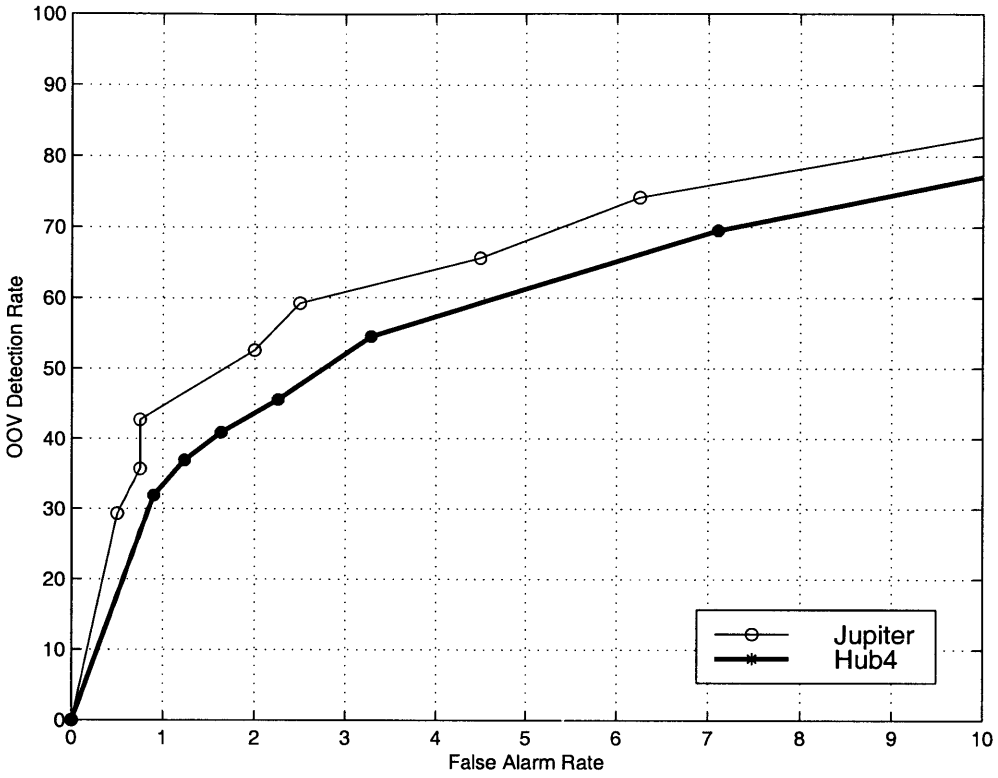

Figure 4-16: ROC curve for OOV detection on JUPITER and HUB4. Both experiments use the same dictionary model to handle OOV words.

4.10 Conclusions

In this section, we reiterate the results and conclusions obtained from all the experiments we reported. The following list gives the highlights of the conclusions and discussions presented throughout the chapter:

1. The OOV approach is capable of detecting half of the OOV words with a very low false alarm rate. Detection can reach up to 70% with a false alarm rate of 5.3%.

2. The impact on IV performance is quite insignificant. IV WER suffers a small increase when the model is introduced into a closed-vocabulary system.

3. The overall WER is reduced when the model is introduced as it allows the correction of errors on words neighboring OOV words. Further improvement in WER is possible if detected OOV words are identified.

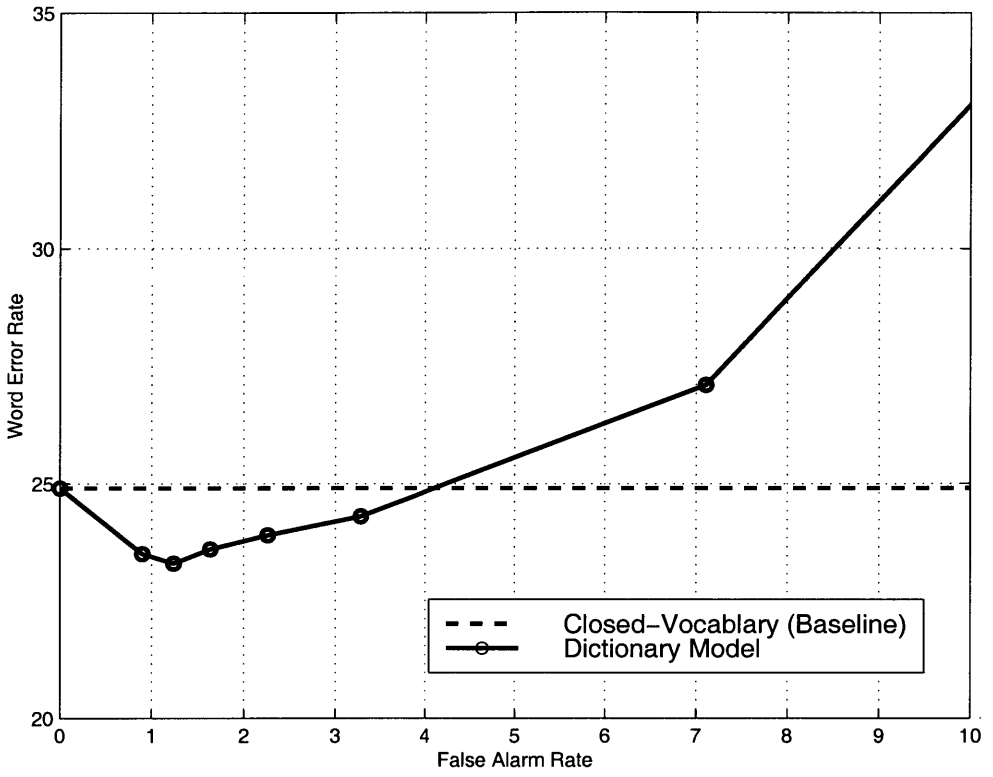

Figure 4-17: WER on the HUB4 test set. Shown is the closed-vocabulary performance (24.9%) and the performance of the dictionary model system as a function of the FAR.

4. The oracle model estimates a conservative upper bound on performance that is superior to other models, indicating that there is room for improvement for this approach.

5. Training the model on a large dictionary of domain-independent words improves performance over just using the training corpus. The FOM improves from 0.54 to 0.64.

6. Imposing topology constraints such as minimum or maximum duration improves performance, but only slightly. We found that a minimum duration of 3 phones performs best.

7. The complement OOV model barely provides any performance gains. This strongly suggests that only novel sequences are considered and generated by the OOV branch.

8. The approach accurately locates most of the detected OOV words. Most of the detected words fall within a window of 20 msec of the true boundaries of the word.

9. With only a phone bigram, the phonetic error rate of detected OOV words is 37.8% for a shift tolerance of 25 msec. Using a second stage and/or higher order n-grams could further improve the phonetic recognition performance.

10. The approach works well in large vocabulary environments. The performance on HUB4 is slightly worse than on JUPITER, but overall WER is reduced by up to 1.4% at best, from 24.9% down to 23.5%.

4.11 Summary

In this chapter we presented an approach for modelling OOV words. The approach relies on creating a hybrid search network by merging an IV search network and an OOV search network. We described how this model can be built within the FST framework. We also presented three different configurations of the OOV model: a corpus model trained from phonetic transcriptions of a training corpus, an oracle model that is intended to measure an upper bound on performance, and a dictionary model trained from a large domain-independent dictionary. We also described two methods to impose hard constraints on the OOV model, including both minimum and maximum length constraints and a complement model that prohibits IV phone sequences within the OOV branch.

The chapter also presented a series of experiments on the JUPITER and HUB4 domains. The goal was to explore the feasibility of the approach presented, and to study its behavior across various configurations. First, we described the four performance measures relevant to the OOV problem including detection quality and recognition. We then described the experimental setup of the two domains and reported on a set of results. In addition, we discussed the impact of applying topology constraints on the model and the performance on large vocabulary domains.

In the following chapter, we extend the approach further to allow for subword units larger than the phone. The two models we presented in this chapter, the corpus and the dictionary models, are based on the standard phone set which is highly unconstrained. Using larger subword units should help to reduce false recognitions of OOV words.

Chapter 5

Learning Units for Domain-Independent OOV Modelling

5.1 Introduction

The models we presented so far for recognizing OOV words rely only on the phone set for subword lexical access. In this chapter, we extend our framework to restrict the OOV network to multi-phone variable-length subword units. First, we start by describing some of the motivations behind the approach. Next, we provide a short literature review on subword recognition covering both knowledge-driven and data-driven approaches. We then describe an information-theoretic approach for learning subword units. Finally, we present a set of experiments for building the subword unit inventory and performing recognition. We compare and contrast the approach in this chapter to approaches we presented so far in the thesis.

5.2 Motivations

Most continuous speech recognition systems model speech as a concatenation of subword units. There are two main reasons for this subword modelling. First, many words share common sub-structures which can be trained jointly. Such sub-structures are quite similar within similar contexts. Second, if systems were to build individual words models, a large

amount of training data is needed for each word. Training common subwords and chaining these subwords together to form complete words can help reduce the amount of training data needed and can allow for a more accurate training of the various phonetic contexts.

There are several subword units that can be used for recognition. The simplest one is the phone, the one we used so far for modelling OOV words. The advantage of using a phone-type inventory of subwords is that it is small and does not require a large amount of data for training. However, the drawback is that performing phone-level recognition is highly unconstrained and phones can undergo severe transformation depending on the context they appear in. A longer but more robust subword unit is the syllable. Syllables typically span several phones within a word, making them easier to recognize during search. Other subword units include automatically derived multi-phone sequences [Deng and Sameti 1994]. Such subword units can span a fraction of a syllable, a whole syllable, or sometimes two or more syllables depending on the way they are derived.

There are two methods in which large subword units (larger than a single phone) are used for speech recognition. In the first method, each subword unit has its own trained acoustic model and words are pronounced in terms of these subword units [Hu et al. 1996; Jones et al. 1997]. In the second method, the words are still pronounced in terms of the subword units, but the units are modelled by simply concatenating the underlying phone sequence of the unit (i.e. no new acoustic models). The second method is usually used for two-stage systems where the first stage allows for subword recognition, while the second stage puts these subwords together to form whole words [Chung 2001].

Because larger subword units provide for better modelling of word sub-structure, we expect such units to provide better structure for modelling OOV words. Before we present our approach for using large subword units, we present a brief literature review on the topic.

5.3 Prior Research

In this section, we review the use of subword units for speech recognition. We break down the discussion into knowledge-driven and data-driven approaches. We review research on subword modelling for speech recognition in general and OOV modelling in particular. We also review work on using subword units for word-spotting.

5.3.1 Knowledge-Driven Approaches

Most of the knowledge-driven approaches for subword recognition use either phones, syllables, or related units as the basic acoustic unit during the search. Some approaches focused on using the syllable as an alternative to the phone. Hu *et al.* [Hu et al. 1996] concatenated phones together to form syllable-like units at locations in the signal where segment boundaries are hard to discern. The results they obtained on a small-vocabulary recognizer are similar to those obtained using a recognizer with phone-like units. Similarly, Jones *et al.* [Jones et al. 1997; Jones 1996] created a syllable-level recognizer for a small vocabulary recognition task and also found that performance is similar to using phonetic units. Other approaches included combining units from different levels together; ranging from phones all the way up to the word level [Pfau et al. 1997]. Using syllables has been shown to be beneficial only when syllables are used in conjunction but not instead of phones. Jones *et al.* [Jones and Woodland 1994] achieved 23% reduction in word error rate on the TIMIT corpus when they applied syllable-level knowledge to the N-best recognizer of a standard recognition framework.

For handling unknown words, Kemp *et al.* [Kemp and Jusek 1996] used a large dictionary with syllable boundary markers and encoded all occurring syllables with a minimal graph that they used during recognition. The use of syllables has also been applied for subword approaches in spoken document retrieval. In [Ng and Zue 1997], Ng *et al.* compared the use of syllables to other subword units for the spoken document retrieval task, where they found that performance with syllables is close to that with other automatically-derived subword units.

5.3.2 Data-Driven Approaches

Data-driven approaches are useful when the linguistic knowledge, such as the syllable boundaries, is not available. Data-driven approaches fall under the general category of automatic language acquisition techniques where the goal is to derive some subword units based on statistics from a large data source.

In [Gorin et al. 1997], Gorin *et al.* performed on-line acquisition of acoustic, lexical and semantic units from spontaneous speech. They used an algorithm for unsupervised learning of acoustic and lexical units from out-of-domain speech data. The new lexical units are

used for fast adaptation of language model probabilities to a new domain. In [Klakow et al. 1999], Klakow *et al.* used automatically generated filler fragments to augment the lexicon for modelling OOV words. The algorithm for deriving the fragments relied on a combination of the frequency and the mutual information of pairs of fragments. In addition, they incorporated these fragments into the word-level language model. Klakow *et al.* used these fragments to reduce the damage on in-vocabulary words, to detect OOV regions, and to provide a phonetic transcription for these regions. The performance of this technique has been evaluated in terms of damage reduction error rate and OOV tagging rate. Significant improvements are reported on both measures when compared to using a simple filler model.

Another interesting approach is the use of automatically-derived multigrams. Multigrams are variable-length phonetic sequences discovered by an iterative unsupervised learning algorithm. This algorithm was originally used for developing multigram language models for speech recognition [Deligne and Bimbot 1995]. Later in [Deligne and Bimbot 1997], the multigram approach was used to derive variable-length acoustic units for speech recognition. In the multigram approach, a subword unit is assumed to be composed of a concatenation of independent variable length phone sequences of a maximum length m. For a given segmentation of a complete sequence into subsequences, the likelihood of the sequence is the product of the likelihood of individual subsequences. The multigrams are derived using maximum likelihood estimation using the iterative expectation maximization algorithm from incomplete data [Dempster et al. 1977].

5.4 The Mutual Information OOV Model

Although the dictionary-based OOV model constrained the W_{OOV} n-gram to model phone sequences in actual words, the topology is still quite simple. Incorporating additional structure into the model should provide for more constraints, and reduce confusability with IV words. One way to incorporate such structure is to use units larger than the phone that can restrict the OOV model to only these subword units.

In designing a subword inventory for OOV modelling, we adopted a data-driven approach. The first reason for choosing a data-driven approach is that it does not require the additional knowledge about the subwords. For example the syllable boundaries are needed to construct a syllable inventory of subwords. The second reason for our choice is to have

control over the inventory size, and hence the size of the OOV model. If we were to use a syllable inventory of words, we will need to include all syllables in the language in order to have complete coverage of all possible words. This could be computationally expensive since the number of syllable is on the order of several thousand distinct syllable (for English). On the other hand, automatically learning the subword inventory allows for more control over the size of the OOV model. For tasks where the OOV problem is severe, one can build a detailed model by using a large inventory of units, while in cases where the problem is not as severe, a small inventory can be devised instead. In the following section, we describe the details of our approach for learning an inventory of multi-phone sequences.

5.4.1 The Approach

Our approach for learning multi-phone sequences relies on a bottom-up clustering algorithm that starts with the basic phoneme inventory and gradually merges pairs of phonemes to form longer multi-phoneme units. In deciding which pairs to merge into a new unit, the algorithm examines phonemes (or unit) co-occurrance statistics of all pairs occurring in a large domain-independent dictionary, and iteratively creates multi-phoneme units which are then used to create the OOV model. The criterion for merging a pair of units is based on the weighted mutual information of the pair. For two units u_1 and u_2, the weighted mutual information $MI_w(u_1, u_2)$ is defined as follows:

$$MI_w(u_1, u_2) = p(u_1, u_2) \log \frac{p(u_1, u_2)}{p(u_1)p(u_2)} \tag{5.1}$$

where $p(u_1)$ and $p(u_2)$ are the marginal probabilities of units u_1 and u_2 in the large dictionary, and $p(u_1, u_2)$ is the probability of u_1 followed by u_2 in the same dictionary. Note that u_1 and u_2 could either be single phonemes, such as at the very first iteration of the algorithm, or multi-phoneme units after some phonemes have been merged. Mutual information measures how much information one unit u_1 contains about the neighboring unit u_2. Note that when the two units are independent, $p(u_1, u_2) = p(u_1)p(u_2)$ and hence $MI_w(u_1, u_2) = 0$, indicating that merging these two units is not very useful. On the other hand, the more dependent the two units are, the higher their mutual information, indicating that such a sequence of phonemes occurs frequently. Since our mutual information is weighted by the joint probability $p(u_1, u_2)$, the frequency of the pair is also represented in

our merging metric.

The mutual information metric has been successfully used for various applications within speech recognition. One common use of mutual information is to learn phrases for variable-length n-gram creation. In [McCandless 1994; McCandless and Glass 1994], the mutual information metric is used to infer multi-word sequences, or phrases for use in language modelling. The approach we are presenting is quite similar to that, except that it is at the phone level and not at the word level.

The iterative process to derive the variable-length units is applied as follows: First, we initialize the unit set to be the same as the phoneme set of the recognizer. At each iteration, we compute the weighted mutual information for all pairs of units that are encountered in the vocabulary. The top M pairs (u_1, u_2) with the largest MI_w are promoted to become new units. Every occurrence of the pair in the vocabulary is replaced with this new unit $u = u_1 u_2$. If this process is iterated for N steps, the number of units generated will be at most $M \times N$. When merging two units u_1 and u_2, if one of the units, say u_1 occurs only within the context of u_2, then u_1 will disappear from the inventory. That is the reason why the number does not grow by exactly $M \times N$. If this procedure is iterated indefinitely, the unit set will converge to the large vocabulary. The number of iterations N was decided empirically and chosen to represent a tradeoff between the complexity of the OOV model and the speed of recognition. The choice of M is related only to computational issues. Ideally, the best choice of M is 1 where only the best scoring pair is merged and promoted to the next iteration. However, merging the top M pairs instead of only the one best speeds up the algorithm by a factor of M.

One byproduct of the iterative process is a complete parse of all words in the large dictionary in terms of the derived units. These parses are used to estimate the OOV model n-gram at the subword unit level. For example, a bigram will require estimating terms of the form $p(u_1|u_2)$. For our work, we used a subword bigram language model.

5.4.2 The Algorithm

Figure 5-1 describes the approach we discussed. Note that the number of occurrences are computed directly from the large dictionary. When two units are merged, they are treated as a single unit in computing the various probabilities. In addition, the constituents of a merged unit are not included in estimating the probabilities for all following iterations.

% constants:
M: the number of units merged at each step
N: the number of iterations the procedure runs

% initial and updated unit inventories
P_{inv}: the initial inventory of phonemes
$U(i)$: the unit inventory at step i

% counts used at each iteration
$C(u, i)$: number of occurrences of unit u in the large dictionary
$C(u_1, u_2, i)$: number of occurrences of pair u_1u_2 in the large dictionary
$C_T(i)$: total count of units at iteration i
$MI_w(u_1u_2, i)$: the mutual information of pair u_1u_2 at iteration i

begin
initialize initialize inventory: $U(0) = P_{inv}$
for $i = 1, \ldots, N$
for all occurring unit pairs $u_1u_2 \in U(i)$
compute $P(u_1) = \frac{C(u_1,i)}{C_T(i)}$
compute $P(u_2) = \frac{C(u_2,i)}{C_T(i)}$
compute $P(u_1, u_2) = \frac{C(u_1,u_2,i)}{C_T(i)}$
compute $MI_w(u_1, u_2, i) = p(u_1, u_2) \log \frac{p(u_1,u_2)}{p(u_1)p(u_2)}$
find the M pairs with highest MI_w
update all occurrences of merged pairs with new units
update $U(i+1) = \{U(i)$; top M pairs $\}$.
end
end
end

Figure 5-1: The algorithm for learning variable-length multi-phoneme subword units. These units are used to construct the OOV model.

5.5 Experiments and Results

In this section, we describe a set of experiments we conducted to derive the subword units using the approach we described above. We first present results on learning the units and discuss some of the characteristics of the iterative procedure. We, then, present the recognition experiments, where we present OOV detection results as well as recognition results. All the experiments for this work are within the JUPITER weather information domain that we described in Chapter 2.

5.5.1 Learning the Subword Inventory

To derive the variable-length units for the OOV model we used the LDC PRONLEX dictionary which contains 90,694 words with 99,202 unique pronunciations. Starting with an initial phoneme set of 62 phonemes, we performed 200 iterations over the unit inventory using the mutual information criterion. For computational reasons, on each iteration we created 10 new units, yielding a total of 1,977 acquired units. Our goal was to acquire an inventory of about 2,000 subword units, roughly the same size as that of the vocabulary of the baseline recognizer. The baseline recognizer has 2,009 words in its vocabulary.

Figure 5-2 plots the mutual information measure for the first 20 iterations. Each curve in the figure corresponds to the ordered mutual information values obtained for all existing unit pairs in a single iteration. Each curve therefore decreases monotonically when plotted against the rank of the ordered values. As one would expect, the top mutual information value (rank 0) decreases with each successive iteration. It is interesting to observe that, on earlier iterations at least, the mutual information values drop off quickly, supporting our heuristic for merging the top 10 pairs on each iteration. One can also see that as the iteration number increases the curves starts to level off. This behavior could possibly be used to develop a more rigorous stopping criterion. As the number of iterations tend to ∞, the curves in Figure 5-2 will converge to a horizontal line. This corresponds to the case when every word in the large dictionary turns into a unit in the inventory.

Tables 5-1 and 5-2 show the top 10 pairs at the start of the procedure and after the 50^{th} iteration, respectively. The first column shows the left constituent of the pair, the second column shows the right constituent, and the third column shows the weighted mutual information MI_w of the pair. Similar tables are provided for the top 100 pairs in Appendix B

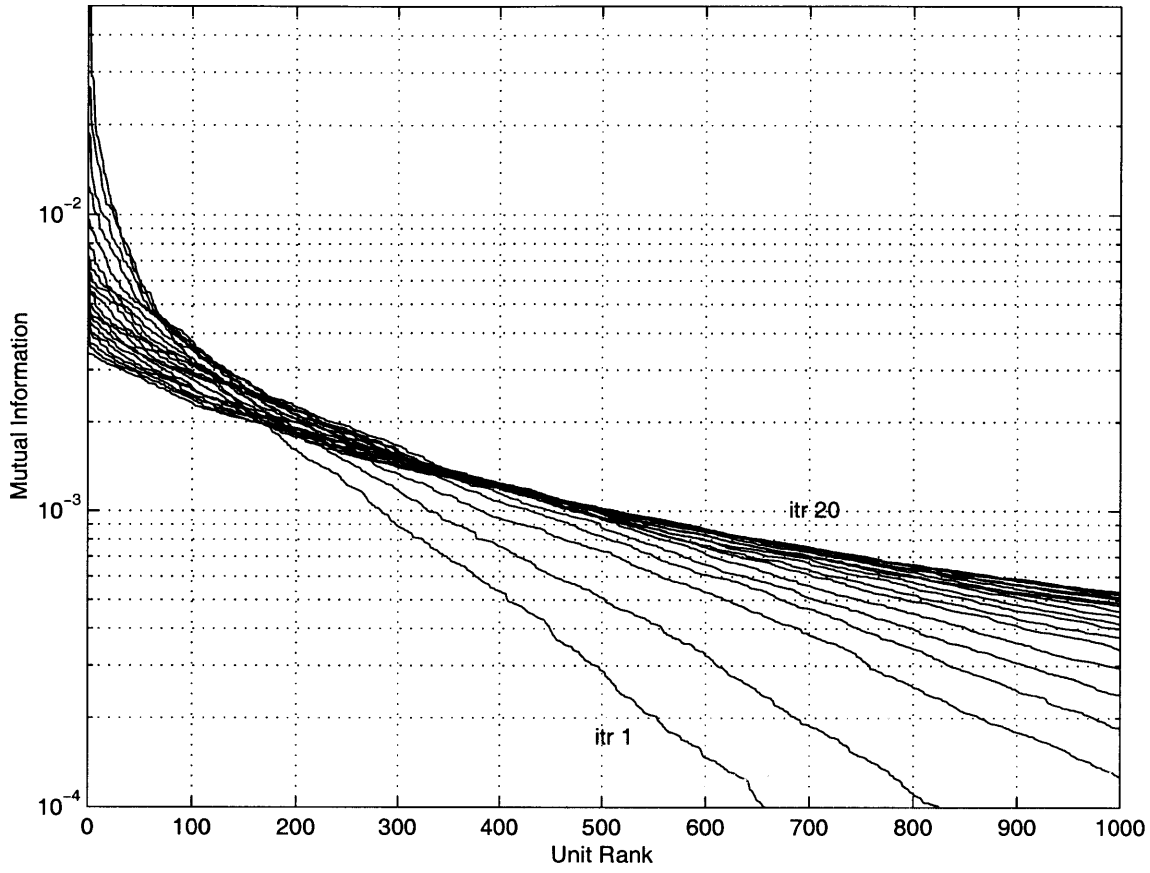

Figure 5-2: Ordered MI values for the first 20 iterations. For each iteration, all pairs of units are sorted in descending based on their weighted mutual information. The weighted mutual information is plotted as a function of the rank of the pair.

while a list of the basic inventory and an example is provided in Appendix A . From the first table, we notice that the top few pairs correspond to pairs of phonemes that tend to occur frequently in that order. For example, the highest ranking pair is *(ix,ng)*; and it occurs in words such as help*ing* and climb*ing*. The second pair is *(ax,n)*; and it occurs in words such as *an*noy and act*on*. In Table 5-2, some of the units are merges of two phonemes from previous iterations, such as the units *er_n* and *r_ax*. The top pair after the 50^{th} iteration is the pair *(b,eh_l)* which occurs in words such as No*bel* and *Bel*gium.

In order to quantify the amount of constraint we were acquiring from the dictionary, we measured phoneme perplexity on the training data of the original 62 phoneme OOV model, the 1,977 unit MI OOV model, and a hypothetical OOV model with 99,202 units constrained by the word baseforms in the training vocabulary. The latter model would have

Left constituent u_1	**Right constituent u_2**	$10^3 MI_w$
ix	ng	83.581
ax	n	63.778
ax	l	29.005
tr	r	28.997
ao	r	28.078
r	iy	22.560
td	s	19.471
s	t-	18.313
tf	ax	18.166
sh	ax	17.585

Table 5-1: The top 10 pairs for the first iteration. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information.

Left constituent u_1	**Right constituent u_2**	$10^3 MI_w$
b	eh_l	1.862
w	ax	1.836
tr_r	ay	1.831
p_r	ay	1.817
t	er_n	1.794
ae	r_ax	1.790
k	ao_l	1.781
d	ow	1.776
eh_nt	ax_l	1.769
k	ao	1.764

Table 5-2: The top 10 pairs after iteration 50. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information.

been attained if the merging procedure had been run until each word became a recognition unit. Of course, this representation would be much larger, and would not generalize to words not in the training vocabulary. Phoneme transitions within a unit had a probability of 1.0, since they were deterministic. As expected, the MI model significantly reduced perplexity of the original OOV model, from 14.04 down to 7.13. The perplexity of the hypothetical model with 99,202 units is 4.36. The fact that the MI OOV model had a significantly lower perplexity than the dictionary OOV model indicates its better ability to predict plausible phoneme sequences and hence could perform better in detecting and recognizing OOV words. Table 5-3 summarizes the results for the three conditions we discussed.

Model Units	**Number of units**	**Perplexity**
Phonemes	62	14.04
MI Units	1,977	7.13
Complete Vocabulary	99,202	4.36

Table 5-3: Model perplexity for three configurations: the phoneme model, the MI model of 1,977 units, and the complete vocabulary model of 99,202 units.

Table 5-4 shows a list of some of the units obtained and the words they represent in the vocabulary. Analyzing the derived units, we observed that around two thirds of the units are legal English syllables as provided in the PRONLEX dictionary. The rest are either syllable fragments or multi-syllable phoneme sequences.

Table 5-5 shows a complete breakdown of the learned units into four groups: legal English syllables, units that contain a single vowel, multi-vowel units, and consonant cluster units. Out of a total of 1,977 learned units, 1,281 are valid English syllables. There are 161 units that contain a single vowel, but are not marked in PRONLEX as legal syllables. In addition, 484 learned units contain more than one vowel, and some of those correspond to frequent word fragment such as *sh_ax_n_ax_l*. Of these 484 multiple-vowel units, 421 units are the result of concatenating two or more legal syllables in PRONLEX. The final group is that of consonant clusters, and it contains 81 units including clusters such as *s_k_r* and *s_tr_r*.

In Figure 5-3, we show the histogram of the length of the derived units (in terms of the number of phonemes). The average length of a derived unit is 3.2 phonemes. The length of a derived unit ranges from a minimum of 1 phoneme to a maximum of 9 phonemes.

Word	Pronunciation
is	ih_z
where	w_eh_r
yugoslavian	y_uw g_ow s_l_aa v_iy ax_n
whisperers	w_ih s p_ax_r axr_z
shortage	sh_ao_r tf_ax jh
unconditional	ah_n k_ax_n d_ih sh_ax_n_ax_l
festival	f eh_s_t ax_v ax_l

Table 5-4: Sample pronunciations with merged units. Some of the units are legal English syllables such as *y_uw*. Others are either syllable fragments such as the phoneme *f* in the word *festival* or multi-syllable units such as *sh_ax_n_ax_l* in the word *unconditional.*

Group	Count	Examples
PRONLEX syllables	1,281	b_ao_r, m_ax_n_z
Units with one vowel	161	ax_g_r, ao_v
Units with multiple vowels	454	sh_ax_n_ax_l, k_aw_nt_axr
Consonant-cluster units	81	s_tr_r, s_k_r, b_r

Table 5-5: Breakdown of the learned units into four types: legal English syllables, one-vowel units, multi-vowel units, and consonant cluster units.

For comparison, Figure 5-4 shows the histogram of the length of legal English syllables in PRONLEX. The average length of a syllable is 3.9 phonemes, longer than that of the MI unit (3.2 phonemes).

5.5.2 Detection Results

To measure detection performance, we use the same performance measure as in the previous chapter. The detection quality of the various OOV models was measured by observing the OOV detection and false alarm rates on the test set as C_{OOV} was varied. Figure 5-5 plots the ROC curves for the four models: the three models we studied in previous chapters (corpus, oracle, dictionary) and the new MI model that is constructed from the learned MI units. Table 5-6 summarizes the FOM measure for the various conditions both for the 0 to 10% range well as for the overall ROC area.

Figure 5-5 shows that overall the MI model performs better than the corpus and dictionary models and approaches the performance of the oracle model. For example, if we wish to operate at a detection rate of 70%, the false alarm rate goes down from 8.5% in the

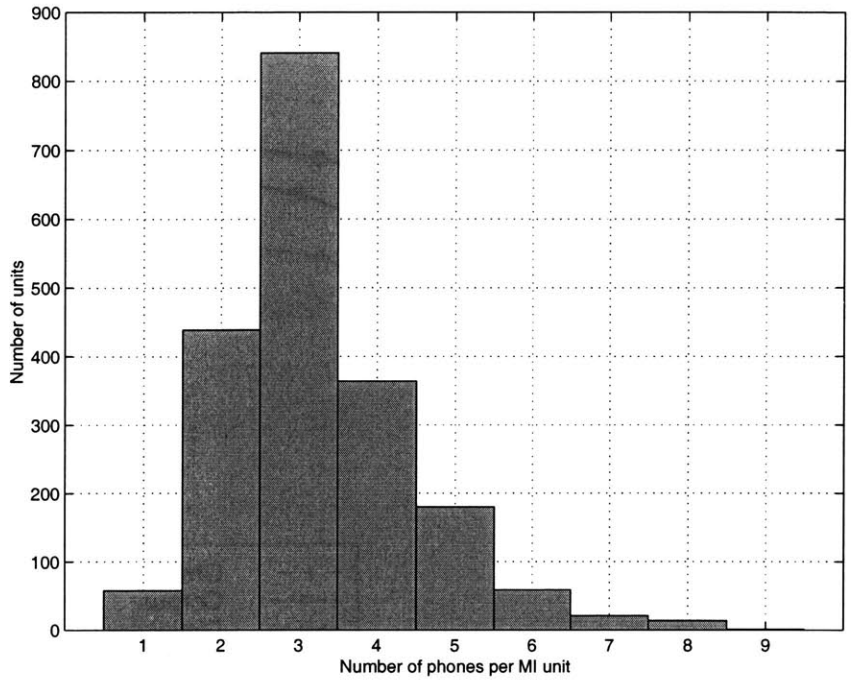

Figure 5-3: Distribution of unit length (in terms number of phonemes). The mean of the distribution is 3.2 phonemes and with a minimum of 1 and a maximum of 9.

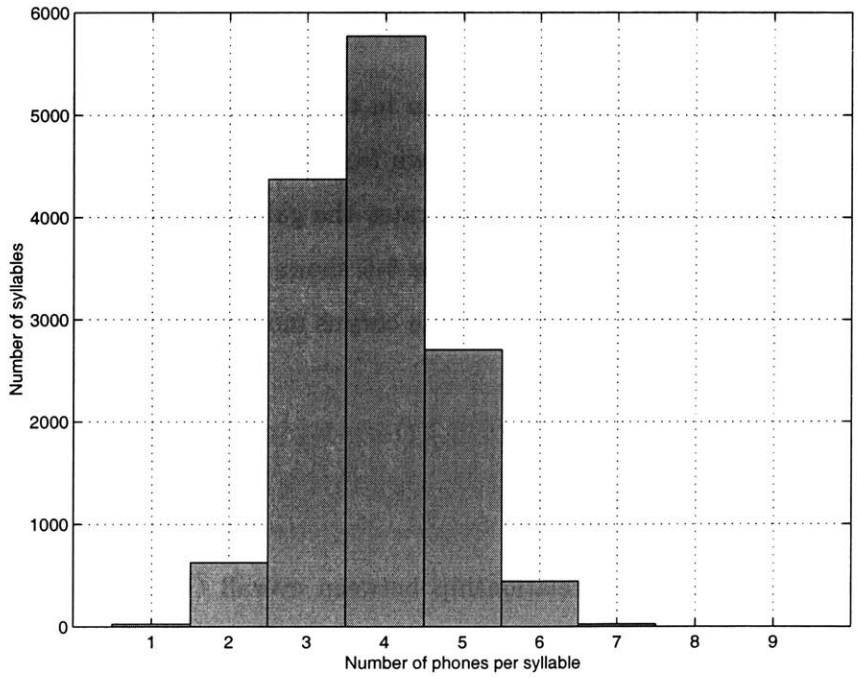

Figure 5-4: Distribution of syllable length (in terms number of phonemes). The mean of the distribution is 3.9 phonemes and with a minimum of 1 and a maximum of 8.

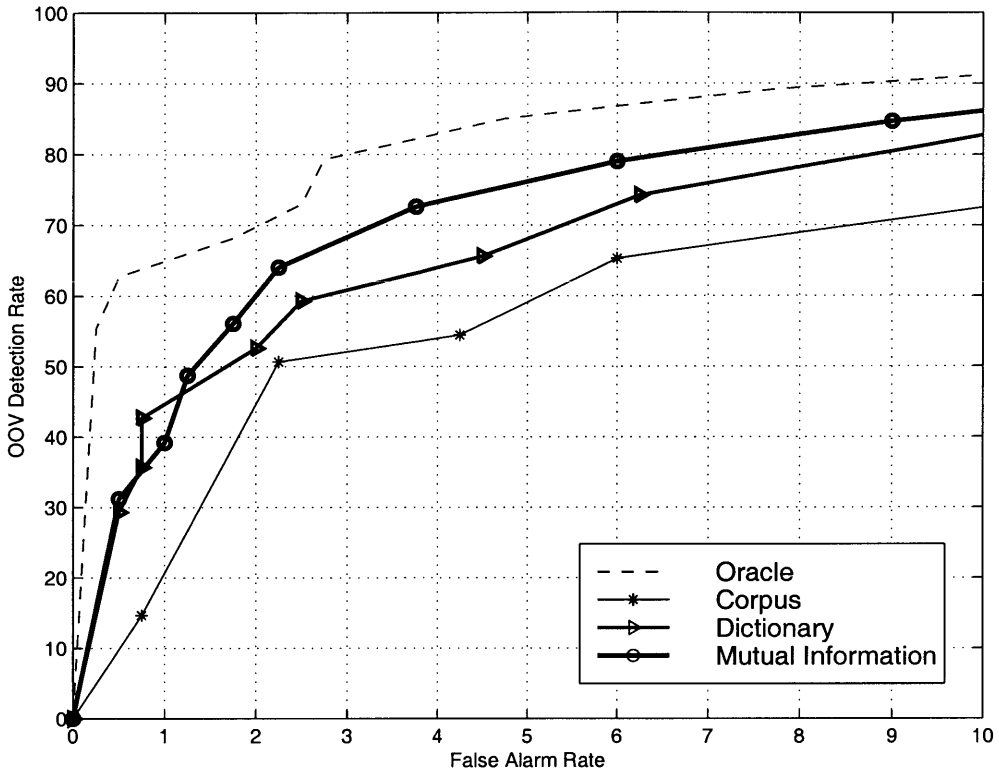

Figure 5-5: ROC curves for the four models: corpus, oracle, dictionary, and MI. Each curve shows the DR versus the FAR for each of the models.

corpus model to 3.2%, i.e., over 60% reduction in the false alarm rate. Compared to the dictionary model, the false alarm rate goes down from 5.3% to 3.2%, about 40% reduction in the false alarm rate. At lower false alarm rates the gain from the MI model is smaller than that at the high false alarm rates. Table 5-6 shows that the overall FOM improves to 0.70, this is an improvement of 30% over the corpus model and 11% over the dictionary model.

5.5.3 Recognition Performance

Similar to previous models, the relationship between overall OOV false alarm rate and word error rate (WER) on IV test data of the MI model is approximately linear. The WER increases slowly from the baseline WER of 10.9% at 0% false alarm rate to 11.6% at 10% false alarm rate. As to the overall WER on the complete test set, the performance is slightly better than the dictionary model. This is illustrated in Figure 5-6. The WER decreases as

OOV Model	100% FOM	10% FOM
Corpus	0.89	0.54
Dictionary	0.93	0.64
Mutual Information	**0.95**	**0.70**
Oracle	0.97	0.80
Random	0.50	0.10

Table 5-6: The figure of merit performance of the four OOV models. The table shows the FOM for the complete ROC curve (100% FOM) as well as for the first 10% of the ROC curve (10% FOM).

we favor the OOV branch more, and then starts to increase again after it hits a minimum of 16.3%.

5.5.4 Phonetic Accuracy

In terms of locating OOV words, the performance of the MI model is similar to that of the dictionary model presented in the previous chapter. Most of the correctly detected OOV words fall within 100 msec of the true boundaries of the word. Table 4-4 shows the phonetic error rates (PER) for the dictionary and mutual information models. The PER is measured on correctly detected OOV words with a shift tolerance window of 25 msec. The PER of 37.8% for the dictionary goes down to 31.2% with the MI model, a relative improvement of 18% in error rate. This improvement is due to the longer subwords units it generates and can be quite useful for applications that try to identify the OOV word from its recognized phonetic sequence.

OOV model	Substitutions	Insertions	Deletions	PER
Dictionary	18.3	12.9	6.0	37.9
Mutual Information	15.2	10.7	5.3	31.2

Table 5-7: Rates of substitution, insertion, and deletion and phonetic error rates (PER) obtained with dictionary model and the MI model for a tolerance shift windows of 25 msec.

To put this result in perspective, we draw a comparison with subword recognition performance reported in [Bazzi and Glass 2000a] where we developed a two-stage recognition system with a subword recognizer in the first stage. This first stage recognizer achieves a PER on the JUPITER domain that ranges from 12% up to 24% depending on the subword

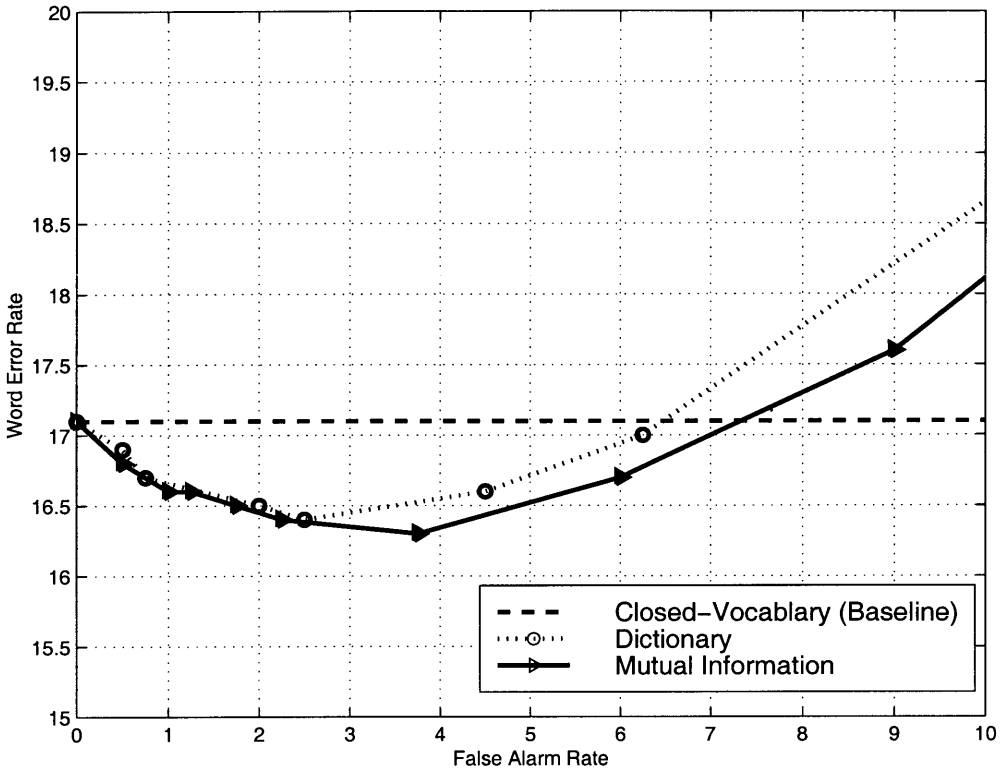

Figure 5-6: WER on the complete test set. Shown is the closed-vocabulary performance (17.1%) and the performance of the dictionary and MI models as a function of the FAR.

unit (phones or syllables) used and the order of the subword n-gram language model. The performance of the MI model with 31.2% PER is quite encouraging since it measures recognition performance on OOV words only. The phonetic recognition results reported in [Bazzi and Glass 2000a] are measured on complete utterances including both IV and OOV words, but most of which are frequent well-trained IV words. Frequent IV words are typically easier to recognizer because of the better trained acoustic and language models for these words. In addition, OOV phonetic recognition may suffer errors at the boundaries due to absorbing or leaving out one or more phones from neighboring words.

5.6 Summary

In this chapter, we presented an OOV model that is based on learning subword units using a mutual information criterion. To construct this model, an inventory of variable-length phoneme sequence units is first learned through an iterative bottom-up procedure that relies

on a mutual information measure. This inventory is then used to construct the OOV model with a bigram language model at the unit level.

We also presented a series of experiments to compare the MI model to models we presented in previous chapters. We found that the inventory of units is mostly legal English syllables. We also presented detection and recognition results that showed that the MI model performs 11% better than the dictionary model and 30% better than the corpus model in detecting OOV words. We also showed that the PER improves by 18% over the dictionary OOV model.

Chapter 6

Multi-Class Modelling for OOV Recognition

6.1 Introduction

The configurations we presented so far for modelling OOV words assume that all unknown words belong to the same class W_{OOV}; and hence are represented using the same model. Instead of using a single word model, we can allow for multiple models of OOV words. Each model could represent a particular category or class of words. For example, a city name model can specialize in recognizing city names, another model can recognize unknown weather terms, and so on. In this chapter, we show how our approach can be extended to model multiple classes of OOV words [Bazzi and Glass 2002]. First, we start by describing the motivation behind this work. Then, we present a short review on multi-class modelling in general and as related to OOV recognition in particular. Next, we present our multi-class approach. Finally, we present experiments for creating multi-class OOV models; and we report results on the JUPITER domain.

6.2 Motivations

There are two main motivations for extending the framework to model multiple classes of OOV words. The first is to better model the contextual and linguistic relationships between the OOV word and its neighboring words. The second motivation is to build more accurate OOV networks specialized to certain categories of words.

A single model for all possible OOV words cannot incorporate much word-level contextual information because it has to cover different classes of words such as nouns, verbs, and adjectives. By pooling words of the same class into their own lexical entry, per-class n-gram parameters can be estimated more reliably. For example, consider the three phrases: *in Boston*, *in Baltimore*, and *in knowing*. With a single model, a bigram will contain only the term $P(in|OOV)$. This clearly should not be the same for city names and for nouns. In a multi-class model, the bigram will contain the terms $P(in|CITY)$ and $P(in|NOUN)$, more accurate estimates for what might follow the word *in*.

At the acoustic level, the goal is to create classes of words such that in each class, words that share the same or similar phone sequences are grouped together and used to train a class-specific subword n-gram language model. For example, words such as *entrapment* and *improvement* may belong to the same class because they end with the same sequence of four phones. Choosing classes and assigning class membership to words are two topics we will discuss throughout this chapter.

6.3 Prior Research

Building multi-class models for handling OOV words is closely related to the topic of class-based language models. First, we review class-based n-grams, then we will look at previous work on class-based modelling of OOV words.

A class-based n-gram language model is an extension to the word n-gram presented in Section 2.2.4, where words are pooled in categories or word classes [Niesler and Woodland 1996]. Class n-grams are intended to improve model generalization and reduce the data sparseness problem. By pooling data for words in the same class, model parameters may be estimated more reliably, and by capturing patterns at the class level as opposed to the word level, it makes it possible to generalize to word sequences not present in the training data. Assuming that a word w_i of a word sequence $W = w_1 w_2 ... w_m$ can belong to one class $c_i = c(w_i)$, the conditional probabilities of the class-based n-gram are written as a product of the word membership score and a class n-gram probability:

$$P(W) = P(c_1)P(w_1|c_1).\prod_{i=2}^{m} P(c_i|c_{i-n+1},\ldots,c_{i-1}).P(w_i|c_i) \tag{6.1}$$

For example, the trigram class probability of a word w given the two previous words uv is:

$$P(w) = P(c(w)|c(u)c(v)).P(w|c(w)) \tag{6.2}$$

The probabilities $P(c_n|c_1 \ldots c_{n-1})$ are estimated in the same way word-based n-grams are estimated. Counts of various classes in various contexts are collected and a maximum-likelihood estimate is then used to estimate the n-gram probabilities. For language modelling purposes, using classes allows for sharing statistics among words of the same class. In addition, it helps reduce the model size by narrowing the contexts to the class level from the word level, thus ameliorating the training data sparseness problem.

In the next two sections, we briefly review the two approaches for determining word classes. The first approach relies on using linguistic knowledge such as the Part-Of-Speech (POS) tags to determine the classes as well as the class membership. The second approach is fully automatic and is based on collecting statistics from a large corpus of training data.

6.3.1 Part-of-Speech Approaches

Part-of-speech tags divide words into classes or categories, based on syntax and how words can be combined to form sentences. POS tags also give information about the semantic content of a word. For example, nouns typically express *things*, and prepositions express relationships between *things*. Most part-of-speech tag sets make use of the same basic categories, such as *noun*, *verb*, *adjective*, and *preposition*. However, tag sets differ both in how finely they divide words into categories; and in how they define their categories. This variation in tag sets is reasonable, since part-of-speech tags are used in different ways for different tasks.

The task of POS tagging is to assign POS tags to words reflecting their POS category. But often, words can belong to different syntactic categories in different contexts. For instance, the string *needs* can have two different tags: in the sentence *he needs to go home*, the word *needs* is a third person singular verb, but in the sentence *he has many needs*, it is a plural noun. A POS tagger should segment a word, determine its possible readings, and assign the right reading given the context.

There are two dominant approaches for POS tagging: rule-based and stochastic. Typical rule-based approaches use contextual information to assign tags to unknown or ambiguous words [Brill 1992; Brants 1997]. In addition to contextual information, many taggers use

morphological information to aid in the disambiguation process. The simplest stochastic taggers disambiguate words based solely on the probability that a word occurs with a particular tag [Geutner 1997]. In other words, the tag encountered most frequently in the training set is the one assigned to an ambiguous instance of that word. An alternative to the word frequency approach is to calculate the probability of a given sequence of tags occurring. This is sometimes referred to as the n-gram approach, pointing to the fact that the best tag for a given word is determined by the probability that it occurs with the n previous tags.

6.3.2 Word Clustering Approaches

Instead of using linguistically-motivated approaches for designing word classes, methods for clustering words into classes as part of an optimization process can be employed. In [Kneser and Peters 1997], an optimization algorithm is used to maximize the training set probability of the language model. Starting with some initial assignment, the algorithm evaluates the change in the training set log probability caused by moving every word in the vocabulary from its current class to every other class. The move resulting in the largest increase in the log probability is selected. This procedure continues until a convergence criterion is met.

In [Brown et al. 1992], a greedy agglomerative clustering algorithm is presented. A bigram language model is assumed and it is shown that for this model the training set log probability can be written as the sum of two terms, the unigram distribution entropy and the average mutual information between adjacent classes. The algorithm initially assigns each word in the training corpus to its own class, and then at each iteration merges that class pair with the least decrease in mutual information. The process is repeated until the desired number of classes has been achieved.

A variation of this approach is presented in [Jardino 1996] where simulated annealing is used to find the optimal class membership. Other methods for clustering words utilized semantic information about the words [Bellegarda et al. 1996; Tamoto and Kawabata 1995] In other approach, words having a similar pattern of transition probabilities are generally clustered into the same word class [Ward and Issar 1996; Farhat et al. 1996]. Due to the large number of words and possible merges at each iteration, considerable care is required during the implementation of such clustering algorithms in order to ensure its computational feasibility.

The only work we are aware of on the use of multi-class models for OOV recognition is the approach presented in [Gallwitz et al. 1996]. In their work, Gallwitz *et al.* constructed five word categories that included cities, regions, and surnames. In addition, they defined a category for rare words that are not in the first five, as well as one for garbage words such as word fragments. In order to estimate the n-gram probability of the various OOV categories, they used an *iterative substitution* approach [Fetter et al. 1996]. In this approach, the vocabulary is iteratively increased in size so that less frequent words are considered OOV words during early iterations and their occurrence contributes to the overall language model probability of the corresponding OOV category. For acoustic modelling, they used very simple models for each of the OOV categories: a flat model that consisted of a fixed number of HMM states with equal probability density functions. They performed experiments on a spoken dialog system that answer questions about German intercity train connections. Their results showed a 6% reduction in WER, from 22.9% to 21.5%, but the OOV detection capability was quite poor. With an OOV rate of 5.3%, they were able to detect only 15% of the OOV words.

6.4 Approach

This section presents a multi-class extension to the basic OOV framework we have been exploring throughout the thesis. This extension allows for modelling multiple classes of OOV words within the same search network. The FST representation for the basic framework is given in Section 4.3 and repeated here with some notational changes:

$$R_{H_1} = C \circ P \circ (L \cup L_u \circ G_u \circ T_u)^* \circ G_1 \tag{6.3}$$

We added the subscript 1 to the word-level language model G_1 to indicate that this grammar models a single class of OOV words. In addition, the hybrid search also has the subscript 1 to indicate the single OOV class. The search space R_{H_1} is constructed based on the union of the two search spaces. The FST union operation, $\cup$, provides the choice of going through either the word network from the vocabulary or through the generic word network. To extend this approach to multiple classes, we can construct multiple generic word models and create a hybrid search network that allows for either going through the IV branch or through any of several OOV branches each representing a class of OOV words.

To describe this approach formally, suppose we have N classes of OOV words, that we are interested in modelling. If we construct N subword search networks, one for each of the N classes, Equation 6.3 becomes:

$$R_{H_N} = C \circ P \circ (L \cup (\bigcup_{i=1}^{N} L_{u_i} \circ G_{u_i} \circ T_{u_i}))^* \circ G_N \tag{6.4}$$

In this formulation, R_{H_N} represents the collection of $N+1$ search networks: the word-level IV search network and N subword search networks, each corresponding to a class of OOV words. The i^{th} subword search network is represented by $L_{u_i} \circ G_{u_i} \circ T_{u_i}$. Here, each of the lexicon L_{u_i}, grammar G_{u_i}, and topology T_{u_i} can either be specific to the network or common to all subword networks. The word level grammar G_N includes the N different classes of OOV words. Similarly, this grammar can either use a class-specific language model probability or can use the same estimate for all classes. In our experiments, we explore various combinations of class-specific networks and word-level grammars.

We explore three different methods to design multiple classes for OOV words. The first approach is based on using POS tagging of a large dictionary. The second approach is a fully automatic procedure for clustering words into classes. The third approach is a combination of the first two. The following three sections describe the details of each of the three approaches.

6.4.1 Part-Of-Speech OOV Classes

Class assignments in terms of POS classifications can be used to design the multi-class OOV model. Starting with a tagged dictionary, words can be broken down into multiple classes. For training the word-level language model G_N, each OOV word in the training corpus is replaced with its POS tag, hence class-specific n-grams can be estimated. Similar to the dictionary OOV model we presented in Section 4.4.2 the subword-level language models, G_{u_i} can be trained on the phone sequences of all words belonging to this class of words.

For modelling OOV words, we use only a small number of POS tags. There are two reasons for using a small number tags. First, many of the POS tags correspond to words that are not typically OOVs, such as function words; those are usually in the vocabulary. The second reason is that with a small number of classes, we can ensure the presence of enough training data for each class. Moreover, in designing OOV classes based on POS

tags, we assume that words can belong to only one class of words. In order to resolve the problem of words belonging to multiple classes, such as words that can be either verbs or nouns, we create intersection classes for POS tags that have significant overlap. For example words that can be either nouns or verbs, such as the word *book*, will belong to the class $(noun \times verb)$.

6.4.2 Automatically-Derived OOV Classes

The second approach we present designing OOV classes is fully automatic and relies on a two-step clustering procedure. Given a list of words, the goal is to break the list down into N lists, one for each of the N classes. The first step is the initialization step. The goal of this first step is to obtain a good initial class assignment for each of the words. The second step is an iterative clustering procedure intended to move words among classes in order to minimize the overall perplexity of the model.

Step 1: Agglomerative Clustering

Step 1 uses agglomerative clustering to arrive at some initial assignment of words to one of N classes. Agglomerative clustering is a bottom-up hierarchical clustering technique that starts by assigning each data point its own cluster. Based on some similarity measure, clusters are successively merged to form larger clusters. The process is repeated until the desired number of clusters is obtained [Duda and Hart 1973].

The procedure uses a similarity measure that is based on the acoustic similarity of words. Given the phone sequences of two words w_i and w_j, the similarity measure $d(w_i, w_j)$ is the *phone-bigram* edit distance between the two words. For example, consider the two words *strong* with pronunciation *"s tr r ao ng"* and *string* with pronunciation *"s tr r ih ng"*. The phone-bigram distance between the two words is $d(strong, string) = 2$ because we would need to change two phone pairs in one to get to the other word. The main reason for choosing such a similarity measure is to ensure some level of similarity not only at the phone level but also at the phone pair level. Because it relies on phone pairs, such a similarity measure is closely correlated with a bigram language model so it should provide for a good initial estimate for Step 2. Given the distance measure between individual words, we use an average similarity measure at the cluster level. At each step of the clustering procedure, we select for merging the pair of clusters X_m and X_n such that the average distance $d_{avg}(X_m, X_n)$ is

minimum. This average distance is computed by taking the average of all distances between every word in cluster X_m and every word in cluster X_n:

$$d_{avg}(X_m, X_n) = \frac{1}{c_m c_n} \sum_{w_i \in X_m} \sum_{w_j \in X_n} d(w_i, w_j) \tag{6.5}$$

where c_m and c_n are the number of words in clusters X_m and X_n respectively. The summation includes all words in each class and computes the phone-bigram edit distance between every word in cluster X_m and every word in cluster X_n. Because of the computational requirement of this type of clustering, we run this step only on a randomly-chosen subset of the large dictionary of words.

Step 2: Perplexity Clustering

Each of the clusters created in Step 1 corresponds to an initial class assignment for the second step. Given these classes, we create a class-specific phone bigram language model for each class. Step 2 follows an iterative optimization technique similar to K-means clustering [Duda and Hart 1973]. The basic idea is to move words from one class to another if such a move will improve the value of some criterion function. For the OOV model, the criterion function we choose is the word's phone sequence perplexity against the various n-grams. In this section, we use the term word perplexity to mean the perplexity of its pronunciation phone sequence against the various n-gram models.

Starting with the initialization from Step 1, the iterative procedure starts by evaluating the perplexity of phone sequence of each word against the N phone language models. At each iteration, each word may move into another class. If the perplexity of the word against the class it belongs to is less than its perplexity against all other classes, the word does not move to another class. On the other hand, if the word perplexity against its class is higher than that against other classes, the word is moved to the class with the minimum perplexity score. The procedure guarantees that at each step, the overall perplexity of multi-class OOV model decreases. The procedure is repeated until the change in average perplexity is smaller than some threshold or no more words change classes. Similar to other iterative optimization techniques, this clustering will converge only to a local minimum and the final class assignment could be quite sensitive to the initial assignment [Duda and Hart 1973].

6.4.3 The Combined Approach

The last approach we present is simply a combination of the first two approaches. Instead of performing the agglomerative clustering to initialize the word classes, the combined approach starts with the assignments from the POS tags and then performs the perplexity clustering described above. There are two advantages for such a combined approach. First, the initial assignment, being based on POS tags, could provide for a better starting point for the perplexity clustering. The second advantage is removing the computational overhead of agglomerative clustering required in step 1 of the second approach.

6.5 Experiments and Results

In this section, we describe a set of experiments to compare and contrast the performance of the various multi-class OOV models against one another as well as against the baseline single class OOV model. We start first by describing results on the POS multi-class model. Then, we move to describing the automatically-derived model and the combined model. All the experiments for this work are within the JUPITER domain that we described in Chapter 3. The training and testing sets are the same as those described in Chapter 4.

6.5.1 The POS Model

For our dictionary OOV model, we used the LDC PRONLEX [McLemore 1997] dictionary which contains 90,694 words with 99,202 unique pronunciations. All words in this dictionary are tagged with one of 22 POS tags in COMLEX. COMLEX is a version of the dictionary that contains a rich set of syntactic feature for use in natural language processing [Grishman et al. 1994]. The majority of the words in PRONLEX belong to one of five main classes: nouns, verbs, adjectives, adverbs, and names. In addition, a significant overlap exists between the noun and verb classes, as well as between the adjective and verb classes. For multi-class OOV modelling, we chose a model with eight classes: the five classes above, the two intersection classes (noun,verb) and (adjective,verb) and a backup class that covers OOV words that are either untagged or don't belong to any of the other seven classes.

Word-Level Perplexity

In order to get some idea of whether the use of eight classes instead of one class improves the word-level language model capability of predicting OOV words, we measured the word-level test set perplexity for IV and OOV test sets. Table 6-1 shows the impact of going from one class to the eight classes on the test set perplexity. Note that the first row of results corresponds to using G_1 while the second row corresponds to using G_8.

Condition	OOV Test Set	IV Test set
1 OOV class (G_1)	25.5	9.6
8 OOV classes (G_8)	23.7	9.6

Table 6-1: Word-level test set perplexity for OOV and IV test sets. The table shows the perplexity using a single class of OOV words, as well as for using the eight POS classes described above.

The table shows the perplexity of the OOV and IV test sets for the two conditions. We make two key observations from these results. First, the IV test set perplexity does not go up with the introduction of the eight classes. Second, the reduction in the OOV test set perplexity is quite significant considering that only the OOV n-grams are the ones that are different between the one class case and the eight class case. This is an indication of the better predictive power of the word-level language model.

Phone-Level Model Perplexity

To build the OOV networks, we use the phone-level lexicon for all eight classes. For each class we train its phone bigram using phone sequences of the words that belong to the class. Table 6-2 shows the model perplexity of the eight different classes, as well as the baseline one class model. The last row in the table shows the weighted average of the eight class model perplexities.

We can see from the table that all classes have lower perplexities than the baseline except for the name class. This can be attributed to more random nature of phone sequences of names. In addition, the weighted average perplexity of the eight-class model is 12.58, 11% lower than the one-class model perplexity, again, indicating the better predictive power of the multi-class OOV network.

Class	Model Perplexity	Number of Words
All words(baseline)	14.04	90,694
adjective	11.61	5,826
adverb	8.55	1,482
name	14.63	22,071
noun	12.96	27,545
verb	10.71	10,665
adjective,verb	7.04	4,927
noun,verb	10.95	5,704
other	13.63	12,474
8 classes (wt.av.)	12.58	–

Table 6-2: Phone-level model perplexity for the baseline single class OOV model, the eight models of the eight classes, and the weighted average of the resulting multi-class OOV model. Also given is the count from PRONLEX of the number of words in each class.

Condition	1 OOV n-gram class (G_1)	8 OOV n-gram classes (G_8)
1 OOV network	0.64	0.65
8 OOV networks	0.68	0.68

Table 6-3: Detection results on different configurations of the POS model. Results are shown in terms of the FOM measure for four different conditions the baseline single class model, and adding multiple classes both at the language model as well as at the OOV network level.

Detection and Recognition Results

Table 6-3 summarizes the detection results for the POS multi-class model. The multi-class extension can be done in one of three ways: (1) only at the language model level by having multiple OOV n-grams, (2) at the OOV model level by having multiple OOV networks, one for each class, (3) both at the language model and the OOV model level. The table shows the three possible cases as well as the baseline. The first result in the table is the baseline system with an FOM of 0.64. The second case involve using the eight OOV classes for language modelling, but still using the same OOV model for all classes, i.e. using G_8 and one OOV network. The FOM for this condition is 0.65, only slightly better than the baseline. The third case involves creating multiple OOV networks but using the same language model n-grams for all classes. The FOM for this case is 0.68, a modest improvement over the baseline single class model. Adding the language model classes to this configuration does

not improve performance. This is the fourth case in the table, where the FOM stays at 0.68. Figure 6-1 shows the ROC curve of this last case.

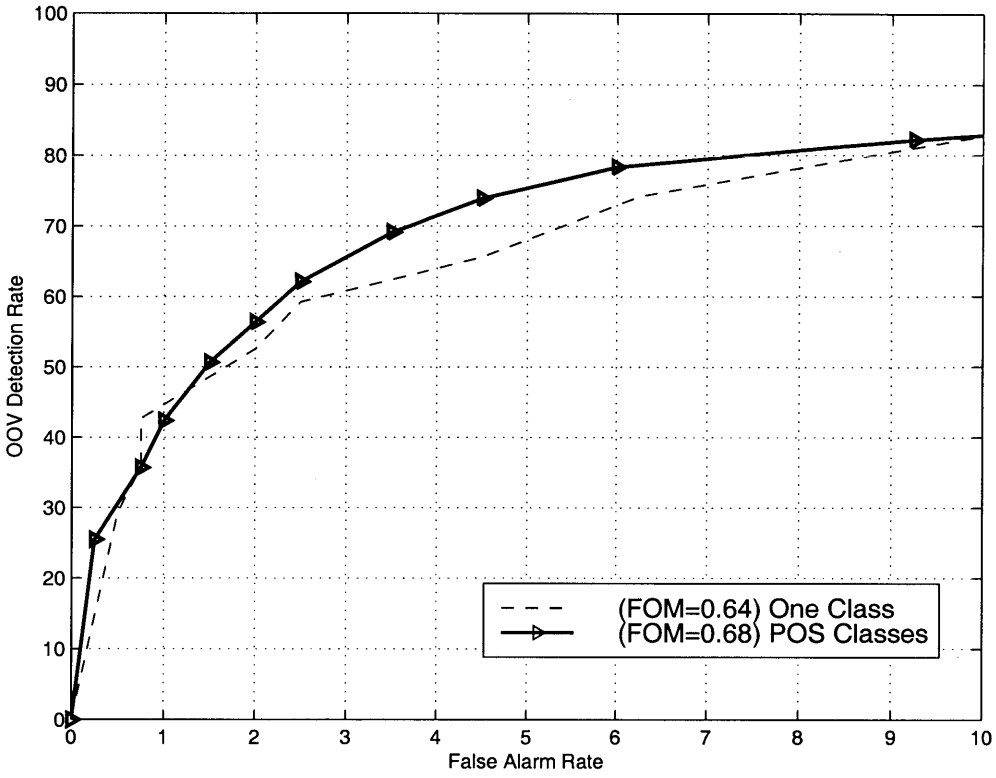

Figure 6-1: ROC plot for the POS multi-class model. Also provided the ROC for the baseline system of a single class model. The ROC shows the case where both the OOV network and the language model are multi-class models.

The FOM results show that the improvement from the POS multi-class model is due mainly to using multiple OOV networks and not multiple language model OOV classes. This finding may be specific to the JUPITER domain since it is a fairly simple recognition task where most of the OOV words are either names or weather terms (nouns). Hence the benefit from the OOV neighboring context is limited and does not help improve performance. Larger domains such as the HUB4 task may benefit more from the multiple language model classes. However, we didn't run multi-class experiments on the HUB4 domain to verify that.

An important aspect of the POS model is its ability to identify the type or POS tag of the OOV word. A manual examination of the correctly detected OOV words showed that 81% of the detected OOV words are recognized with the correct POS tag. The impact of the multi-class model on the WER is similar to that of a single class model. The IV WER

degrades slightly as we allow for more OOV detection, while the overall WER decreases for low false alarm rates and then increases as false alarm rates increases. Behavior is similar to that shown in Figures 4-11 and 4-12.

6.5.2 The Automatically-Derived Model

In this section, we present the detection results for the fully automatic approach as well as the results of the combined approach. In Step 1, agglomerative clustering was done on a 1,000 randomly chosen subset of the PRONLEX word list. We chose $N = 8$ classes in order to compare results with the POS model. The final eight clusters of words are then used to train the phone n-grams for each of the classes. These clusters are then used to seed the perplexity clustering in Step 2. Figure 6-2 shows the change in the weighted average perplexity of the multi-class model in terms of the iteration number for both two initial conditions: the clusters obtained from agglomerative clustering in Step 1, and the clusters based on the POS tags. The clustering was iterated until the change in perplexity was less than 0.05. At that point, very few words moved from one class to another. Figure 6-2 shows that the multi-class model perplexity improves from 12.5 down to 10.2 for the POS initialization and from 13.2 down to 10.3 for the agglomerative clustering initialization.

OOV Model	Number of Classes	FOM
Corpus	1	0.54
Dictionary	1	0.64
Mutual Information	1	0.70
POS	**8**	**0.68**
PP Clus (AggClus Init)	**8**	**0.71**
PP Clus (POS Init)	**8**	**0.72**
Oracle	-	0.80
Random	-	0.10

Table 6-4: The figure of merit performance of all OOV models we explored. The ones in bold face are the multi-class FOMs.

As to the language model, the multi-class model we create with this approach relies on a single OOV class. There are two reasons for not going with multiple classes. The first is the fact that we did not get much gain from multiple language model OOV classes with the POS model. The second reason is that these automatically derived classes may

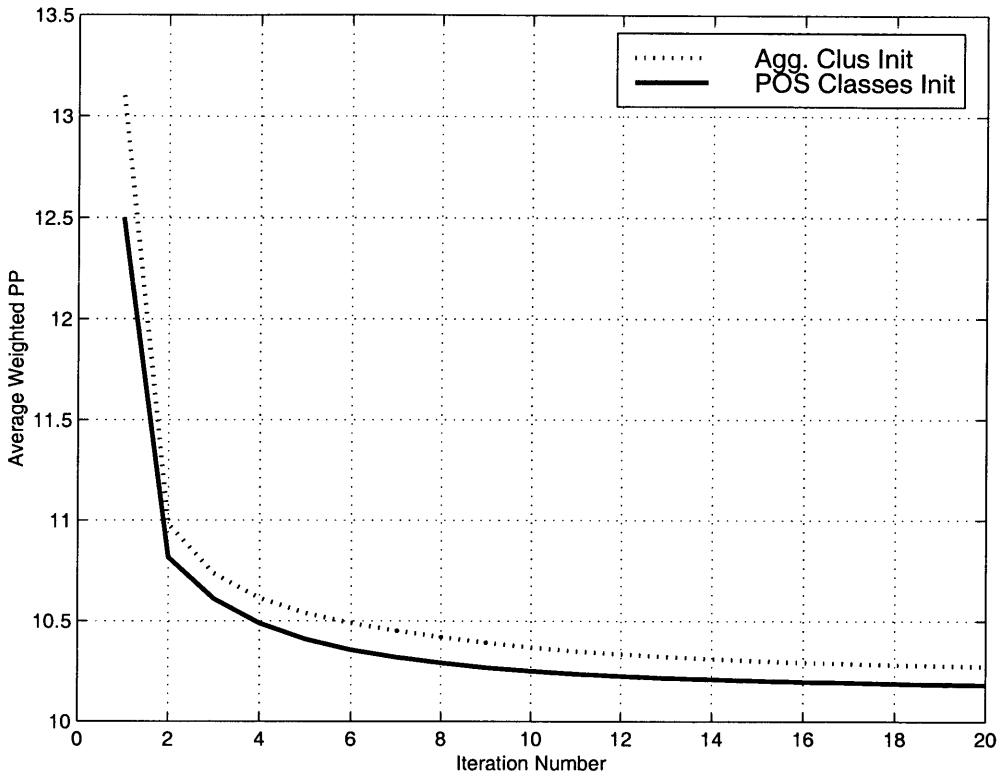

Figure 6-2: Weighted average perplexity of the multi-class model in terms of the clustering iteration number. Shown are the cases for an agglomerative clustering starting initialization and a POS tag initialization.

be acoustically similar because of the way they are derived. However, the words in each cluster do not necessarily share similar contexts.

Detection results are summarized in Table 6-4 and Figure 6-3. As shown, the automatic OOV model with the POS initialization outperforms the single class model as well the POS model. Note that using POS tags for initialization is only slightly better than using agglomerative clustering. Over the baseline system of using one class, the multi-class model improves the FOM by over 11% (from 0.64 to 0.72). For comparison purposes, we also provide the ROC of the three systems: dictionary, mutual information, and automatic multi-class in Figure 6-4.

Varying the Number of Classes

For the results we presented so far, we used eight classes for the automatic multi-class model in order to have a fair comparison with the POS model. However, with agglomerative

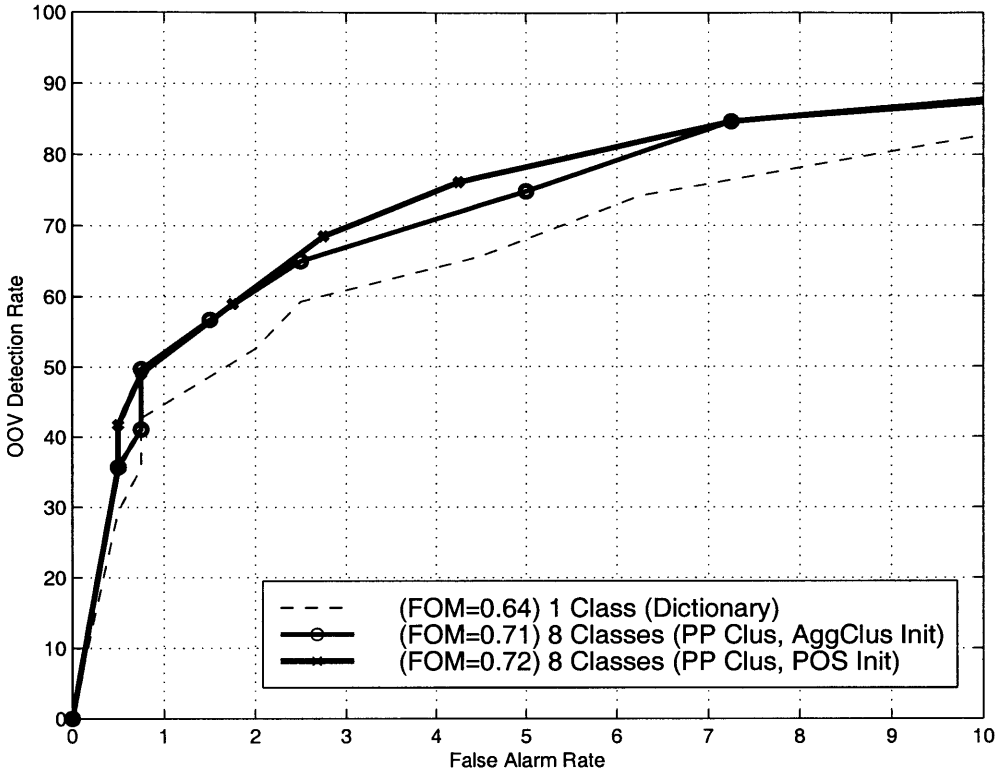

Figure 6-3: ROC plot for the two automatic multi-class models. Also provided the ROC for the baseline system of a single class model. Plots for both agglomerative clustering and POS initializations are shown.

clustering, we can stop at any number of classes and build as many OOV models. Figure 6-5 shows the performance of the multi-class model as a function of the number of classes N. We tried 1, 2, 4, 8, 16, and 32 classes as shown in the figure.

Figure 6-5 shows that most of the gain comes in going from one to two classes where the FOM jumps from 0.64 to over 0.69. The benefit from using more classes diminishes as N increases. Going from eight to sixteen and then to 32 classes gives only a slight improvement in the FOM. An important point to make here is that this behavior could be specific to JUPITER, and other unconstrained domains may benefit more from a larger number of classes.

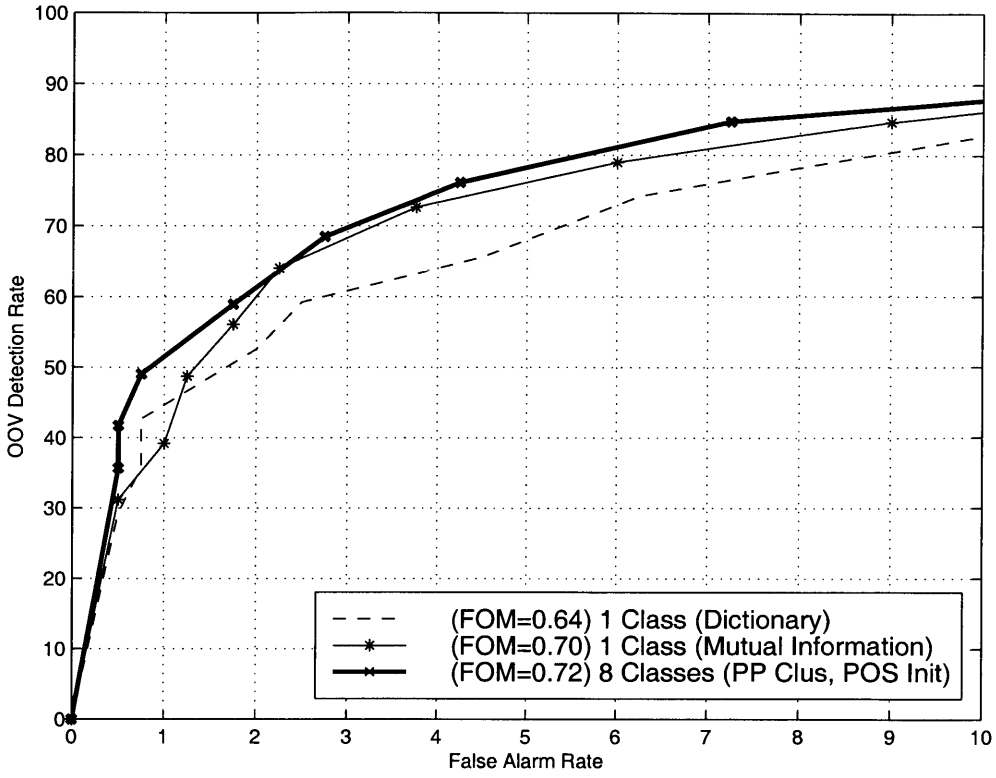

Figure 6-4: ROC plots for the three systems: dictionary, mutual information, and automatic multi-class.

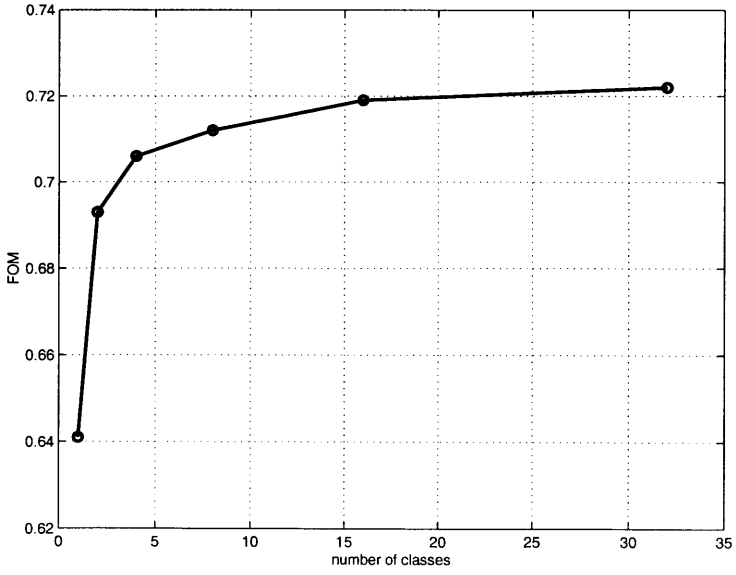

Figure 6-5: The FOM performance of the automatic model as a function of the number of classes.

6.6 Summary

In this chapter, we presented a multi-class extension to our approach for modelling OOV words. Instead of augmenting the word search network with a single OOV model, we add several OOV models, each corresponding to a class or category of OOV words. We presented two approaches for designing the OOV classes. The first approach is knowledge-driven and relies on using common POS tags to design the OOV classes. The second approach is data-driven and devises a two-step process. The first step is a bottom-up hierarchical clustering used to derive an initial class assignment. The second step is an iterative clustering algorithm that tries to move words from one class to another to minimize the overall perplexity of the model. We also presented an approach that combined the first two approaches.

We also presented a series of experiments to compare and contrast the two approaches as well as the multi-class to the single class approach. We found that using the POS multi-class gives an FOM improvement of 6% over the single class model. The automatically-derived model improves FOM by 11%. The combination of the two approaches yields results which are only slightly better than the fully automatic approach with 13% FOM improvement over the single class model.

Chapter 7

Combining OOV Modelling and Confidence Scoring

7.1 Introduction

Recognition confidence scoring is the process of evaluating the reliability of the recognition hypothesis. This is typically accomplished by extracting various measurements from the recognizer about the quality of the recognition output. Such measurements can then be used by a second stage system to enhance overall system performance. One common example is a dialog system that utilizes confidence scores to decide on the next course of action. A low confidence score may cause the dialog system to confirm with the user before taking the action, while a high confidence score could signal the dialog system to go ahead and execute the action.

Confidence scoring and OOV modelling are two similar techniques in the sense that they both try to detect the presence of recognition errors. Both techniques attempt to identify the regions of an utterance where the recognizer cannot find reliable word hypotheses without harming the regions where the recognizer is performing correctly. However, the modelling approaches are quite different. OOV modelling involves modifying the way recognition is performed by explicitly introducing an OOV model during the search in order to identify potential unknown words during recognition, while with confidence scoring, the recognizer's hypotheses are typically post-processed with a confidence scoring model in order to identify hypothesized words which may be misrecognized. Because of the inherently different nature

of the two techniques, they have different advantages and disadvantages, and a combination of the two might prove beneficial.

In this chapter, we compare and contrast the two techniques both in their ability at detecting OOV words as well as in detecting recognition errors. We also present a method for combining the techniques and provide experimental results demonstrating the performance gains that can be obtained by the combined approach. First, we provide a brief overview on confidence scoring and its use for OOV detection and recognition. We also describe the confidence scoring approach within the SUMMIT system. Then we present the approach for combining confidence scoring with OOV modelling. Finally, we describe the set of experiments to compare and combine the two techniques.

7.2 Prior Research

As the value of confidence scoring for speech recognition has been obvious for quite some time, much work has been done in the field [Rivlin et al. 1996; Gillick et al. 1997; Chase 1997a]. Most approaches are based on finding some key features indicative of a correct recognition or understanding and then deriving a confidence metric from them. The main differences among various approaches are accounted for the specific set of confidence features proposed, as well as the methods with which the features are combined. The number and choice of features for confidence scoring is usually very large. Some of the most commonly used features include measurements extracted from the acoustic scores at the phone, word, and utterance level. Other set commonly used features are those that are based on language model information and N-best list statistics, such as the number of times and the locations that a particular word appears in the list.

After all these features are extracted, a mechanism for combining all of them into a single metric is required. Various methods have been used including a simple Bayesian formulation of a two-way classification problem, i.e., whether the recognizer hypothesis is correct or not [Siu et al. 1997; Pao et al. 1998; Kamppari and Hazen 2000]. The classification can be done at the utterance level to determine whether to accept or reject the whole sentence. It can also be done at the word level to decide whether that word should be accepted or rejected. Other methods include using a neural network to combine features such as the work in [Weintraub et al. 1997; Williams and Renals 1997]. The input

to the neural network are the various proposed features and the output is a decision on whether to accept or reject the recognizer hypothesis.

7.3 Confidence Scoring in SUMMIT

This section describes the confidence scoring framework in the SUMMIT system. Details of this framework are described in [Kamppari and Hazen 2000; Hazen et al. 2000a; Hazen et al. 2000b]. In SUMMIT, confidence scores are computed based on a set of confidence measures extracted from the computations performed during the recognition process [Chase 1997b]. For each recognition hypothesis, a set of confidence measures are computed and combined together into a confidence feature vector. The feature vectors for each particular hypothesis are then passed through a confidence scoring model which produces a single confidence score based on the entire feature vector. This score can then be evaluated by an accept/reject classifier which produces an accept/reject decision for the hypothesis. Both utterance level and word level confidence scores are used within SUMMIT.

For each hypothesized word, a set of word level features are extracted from the recognizer to create a confidence feature vector. For this work 10 different features, which have been observed to provide information about the correctness of a word hypothesis, were utilized. These features are:

1. **Mean Acoustic Score:** The mean log likelihood acoustic score across all acoustic observations in the word hypothesis.

2. **Mean Acoustic Likelihood Score:** The mean of the acoustic likelihood scores (not the *log* scores) across all acoustic observations in the word hypothesis.

3. **Minimum Acoustic Score:** The minimum log likelihood score across all acoustic observations in the word hypothesis.

4. **Acoustic Score Standard Deviation:** The standard deviation of the log likelihood acoustic scores across all acoustic observations in the word hypothesis.

5. **Mean Difference From Maximum Score:** The average difference, across all acoustic observations in the word hypothesis, between the acoustic score of a hypothesized phonetic unit and the acoustic score of highest scoring phonetic unit for the same observation.

6. **Mean Catch-All Score:** Mean score of the catch-all model across all observations in the word hypothesis.

7. **Number of Acoustic Observations:** The number of acoustic observations within the word hypothesis.

8. **N-best Purity:** The fraction of the N-best hypotheses in which the hypothesized word appears in the same position in the utterance.

9. **Number of N-best:** The number of sentence level N-best hypotheses generated by the recognizer.

10. **Utterance Score:** The utterance confidence score generated from the utterance-level features. Utterance level features are described in Appendix C.

The feature vector for each individual word hypothesis is then evaluated using a confidence scoring model which produces a single confidence score based on the entire feature vector. To produce a confidence score for a word from the confidence feature vector, a simple linear discrimination projection vector is trained. This projection vector reduces the multi-dimensional confidence feature vector for the hypothesis down to a single confidence score. Mathematically this is expressed as

$$c = \vec{p}^{\,\mathrm{T}} \vec{f} \tag{7.1}$$

where $\vec{f}$ is the feature vector, $\vec{p}$ is the projection vector, and c is the raw confidence score. A threshold on this score can be set to produce an accept/reject decision for the word hypothesis. In our experiments, this threshold is varied to adjust the balance between false acceptances of misrecognized words and false rejections of correctly recognized words. This score is the converted into a probabilistic score which can be used in later processing by the language understanding and dialogue components of the system [Hazen et al. 2000b].

The projection vector $\vec{p}$ is trained using a *minimum classification error* (MCE) training technique. In this technique the projection vector $\vec{p}$ is first initialized using Fisher linear discriminant analysis. After the initialization of $\vec{p}$, a simple hill-climbing MCE algorithm iterates through each dimension in $\vec{p}$ adjusting its values to minimize the accept/reject classification error rate on the training data. The optimization continues until a local

minimum in error rate is achieved. Though this discriminatively trained projection vector approach is quite simple, it has performed quite well in detecting recognition errors [Hazen et al. 2000b].

7.4 Combining OOV Detection and Confidence Scoring

Combining multiple estimators to obtain a more accurate final result is a well-known technique in statistics. In the domain of speech recognition, it has been discovered that combining the outputs of different classifiers and/or recognizers can improve recognition accuracy and robustness [Halberstadt 1998]. A collection of recognizers with different characteristics, such as the use of different input representations, subword models or statistical modelling techniques, offers the potential of extracting complementary information from the speech signal, thus enabling recognizers to compensate for each other's errors. It should come as no surprise that combining systems can lead to performance improvements, but it is more impressive how very significant these improvements can be [Fiscus 1997]. This implies that the different features and approaches extract important information that is being missed by their peers.

Results from combining different speech recognizers are most compelling when the different recognizers utilize different observation measurements or modelling approaches but achieve similar results. Under these circumstances, the expected gain from combining the different classifiers is the greatest. This was the motivation for attempting to combine the two distinctly different methods for detecting recognition errors: OOV modelling and confidence scoring. Our OOV word modelling approach operates during the recognition search process by allowing the recognizer itself to hypothesize a generic OOV word model as an alternative to a known word. On the other hand, our confidence scoring approach is applied as a post-processing technique after the recognition search is already complete.

7.4.1 Approach

A natural way to combine both methods is to enable OOV word detection during recognition and then utilize confidence scoring on the hypothesized known words (excluding the OOV word hypotheses) after recognition is complete[1]. So detecting errors happens in two stages:

[1]This work draws from joint work with T.J. Hazen reported in [Hazen and Bazzi 2001]

the first stage involves *always* rejecting every OOV word hypothesis. The second stage examines the confidence scores of the remaining words in the hypothesized utterance. If the score is above some threshold, the word is accepted, otherwise it is rejected.

For example, consider the input and the recognition results for an utterance given in Table 7-1. The utterance contains the OOV word *Franklin*. Furthermore assume that the confidence scores are normalized such that the accept/reject threshold is set to zero. From the table, we can see that the recognizer detects the presence of the OOV word and makes one mistake with the last word in the utterance. For such an utterance, we first reject the OOV word hypothesis based on the fact that this could be word that we do not know. Second, we examine the confidence score of the remaining words. In this example the word *today* will be rejected because its score is negative.

Input:	what	is	the	weather	like	in	Franklin	for	tomorrow
Hypothesis:	what	is	the	weather	like	in	<OOV>	for	today
confidences:	5.32	6.32	4.56	5.48	4.33	6.61	-0.4	5.57	-1.53

Table 7-1: An example utterance with an OOV word *Franklin*. Shown the input, output of the recognizer and the word-level confidence scores.

Using this two-stage approach there are two opportunities for the system to detect potential errors. During the recognition stage the OOV word detection approach replaces potential misrecognitions with unknown word markers. In the post processing stage, the confidence scoring module examines the remaining word hypotheses which are in-vocabulary and rejects those word hypotheses in which it has low confidence.

7.5 Experiments and Results

7.5.1 Experimental Setup

Experiments presented here utilize the recognizer for the JUPITER weather information domain described in Chapter 4. The word lexicon consists of a total of 2,009 words, many of which have multiple pronunciations. Word class trigram language models are used at the word-level. The training set used for these experiments consists of 88,755 utterances used to train both the acoustic and the language models. The test set consists of 2,388 spontaneous utterances collected by JUPITER, 13% of which contain OOV words and an OOV rate of

2.2% at the word level. On this test set the baseline recognizer has a word error rate of 21.6%. The OOV model we use for these experiments is the *corpus* model presented in Chapter 2. These experiments were conducted before we developed the better performing OOV models (dictionary, mutual information and multi-class models), the reason why we use the corpus OOV model.

In our experiments, we first examine the capability of the OOV detection method and the confidence scoring method on the task of detecting errors caused by unknown words. Second we compare the two methods on the task of detecting recognition errors in general. Finally, we examine the method for combining the two approaches on the task of keyword recognition error detection.

7.5.2 Detecting OOV Words

The purpose of the OOV word detection model is to detect the presence of OOV words without harming the recognition accuracy on correctly recognized known words. Similarly, it is hoped that the confidence scoring module will reject word hypotheses when the actual word is an unknown word without absorbing false rejections of correctly recognized known words. The performance of the two methods on the task of OOV word detection is shown in Figure 7-1. In this figure OOV word detection (i.e., the rejection of word hypothesis errors caused by unknown words) is plotted against the false rejection rate of correctly recognized words. As can be seen in the figure the OOV detection method performs better at the task of detecting errors caused by OOV words than the confidence scoring method. For example, at 70% detection rate, the OOV model has 2.8% false rejection rate, less than half the false rejection rate of the confidence scoring method of 5.8%.

It is not surprising that the OOV method outperforms the confidence scoring method considering that the OOV detection method is designed specifically for this task while the confidence scoring method is designed for the more general task of detecting *any* type of recognition error (including substitution of known words and insertions).

7.5.3 Detecting Recognition Errors

As mentioned earlier, the confidence scoring model is designed to be a generic detector of recognition errors. Its focus is not specifically on the detection of errors caused by unknown words, as examined in the previous section. To test this capability, we can examine the

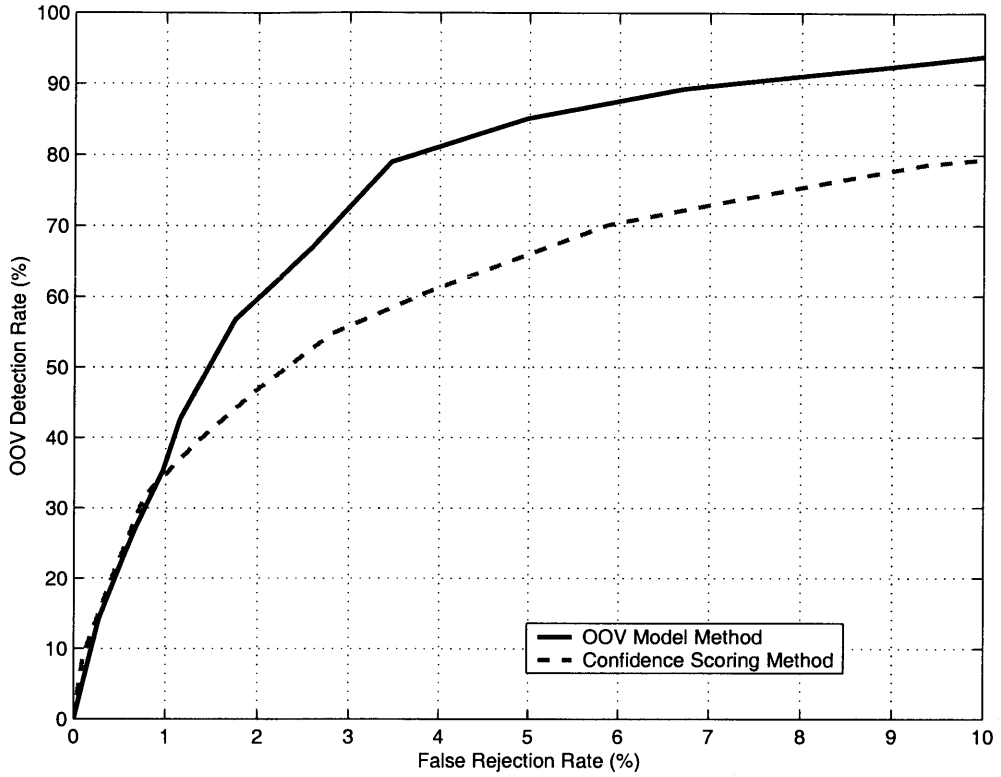

Figure 7-1: Comparison of the rejection rate of errors caused by OOV words versus the false rejection rate of correctly recognized words.

ROC curve of the system. This ROC curve, unlike the one we have seen so far, measures the relationship between the percentage of correctly recognized words which are accepted (i.e., the correct acceptance rate) against the percentage of incorrectly recognized word which are accepted (i.e., the false acceptance rate). Ideally we would like to minimize the false acceptance rate without harming the correct acceptance rate.

Figure 7-2 shows four different ROC curves. The solid ROC curves show the OOV detection method and the confidence scoring method when applied to all words hypothesized by the recognizer. These lines indicate that the confidence scoring method has a better ROC curve than the OOV detection method when applied to all hypothesized words. This result is not surprising considering that the confidence scoring method was specifically designed for this task, while the OOV detection method was designed specifically for detecting errors caused by OOV words.

However, the dashed lines in Figure 7-2 show the ROC curves for the two methods when only examining certain keywords which are important for correct understanding. These key-

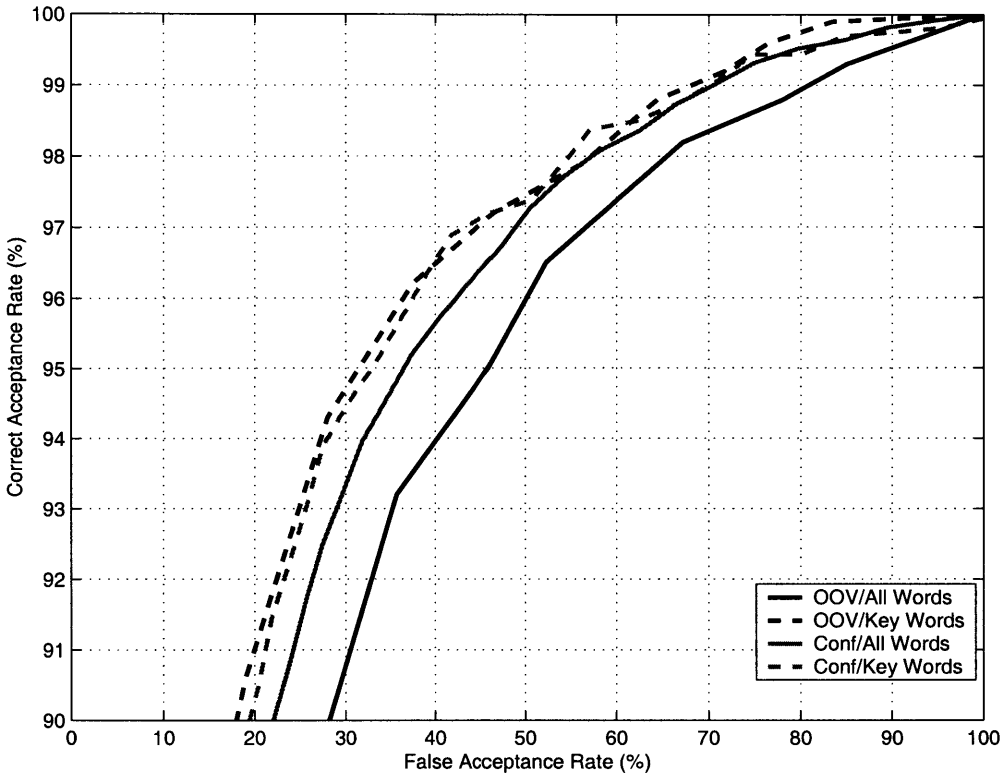

Figure 7-2: ROC curves for OOV word detection and confidence scoring methods evaluated on all words and on keywords only.

words are from a list of 937 proper names of geographic locations known by the recognizer. For this test the two methods perform almost identically. The fact that the OOV detection method works much better on this keyword evaluation than it did on the evaluation using all words is also not surprising. Many of the OOV words that appear in the JUPITER task are proper names of locations. Because of language modelling constraints it is relatively common for the baseline recognizer to substitute a known location for an unknown one.

7.5.4 The Combined Approach

Figure 7-3 shows the ROC curves for keyword detection for our original two methods plus the combined method. In the combined approach, the OOV modelling component is first fixed at a particular operating point before confidence scoring is utilized. The dashed and dotted line on the figure shows one example ROC curve for the combined approach for one initial OOV modelling operating point. In this example, the initial OOV modelling operating point is fixed at a correct acceptance rate of 99.2% with a false acceptance rate of

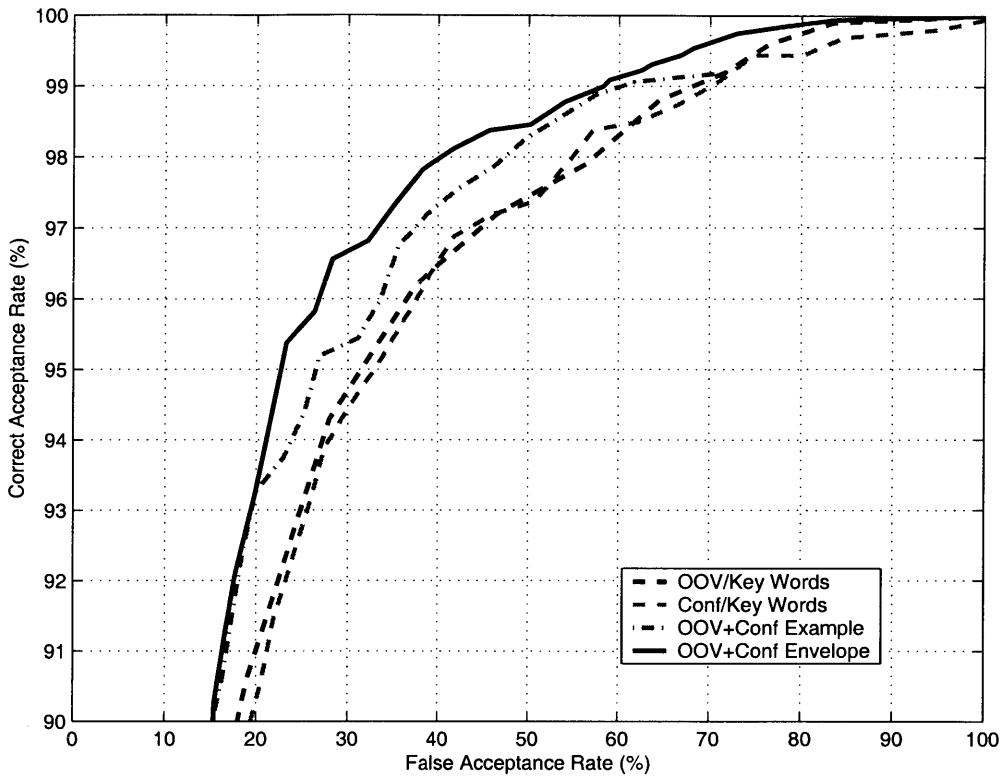

Figure 7-3: ROC curves on hypothesized keywords only using the OOV word detection and confidence scoring methods as well as a combined approach.

71%. From this point we can then generate the remainder of the ROC curve by adjusting the confidence score accept/reject threshold. The solid line shows the optimal ROC curve for the combined approach which is generated by sampling results from all combinations of all operating points for the two different methods and extracting the "envelope" of these dual operating points. These curves demonstrate the significant improvement that can be obtained by the combined approach.

To further illustrate the improvement that can be obtained, suppose we wish to operate our system at a correct acceptance rate on keywords of 98%. At this operating point, the figure shows that the combined approach can reduce the false acceptance rate of misrecognized keywords by over 25% from either of the two original methods (from 55% to 40%). Although not shown, similar (though smaller) improvements can also be observed on the more general task of identifying recognition errors across all words. However, it is important to note that we focused on the keywords because they are most relevant to the overall understanding rate of the full system.

7.6 Summary

In this chapter we presented a method for combining OOV detection with confidence scoring for the task of detecting recognition errors. The first technique uses an explicit OOV model for detecting OOV words. The second approach relies on a confidence model to predict recognition errors. The combined approach uses OOV detection in a first stage and then confidence scoring in a second stage.

In experiments comparing the two techniques, we found that the OOV modelling approach does better at detecting OOV words while the confidence scoring approach performs better in detecting misrecognitions in general. However, the combined approach shows significant improvement over either of the two approaches, especially in detecting recognition errors for domain keywords.

Chapter 8

Summary and Future Directions

8.1 Summary

This thesis addressed the problem of handling OOV words in continuous speech recognition. The significance of the OOV problem is related to how often it occurs and to its impact on system's performance. Because speech recognizer operate on a closed set vocabulary of words, the presence of OOV words is inevitable. However, the OOV rate varies from one task to another. When an OOV word occurs, recognition performance degrades significantly as errors may spread into neighboring words. In this section we summarize the findings and contributions of this thesis.

A survey of existing approaches

A survey of current and previous work showed that approaches to the OOV problem can be classified into four categories. The first category is vocabulary optimization where words are chosen in such a way to reduce the OOV rate by as much as possible. Vocabulary optimization can not totally eliminate the OOV problem because words form an open set and there will always be new words encountered during recognition. Another drawback of vocabulary optimization is that increasing the vocabulary size makes the recognition more computationally expensive and could degrade performance due to larger number of words to choose from during recognition. The second strategy is the use of confidence scoring to predict whether a recognized word is actually a substitution of an OOV word. The main weakness of this strategy is that such confidence measure are good at predicting whether a hypothesized word is correct or not, but unable to tease apart errors due to OOV

words from those errors due to other phenomena such as degraded acoustic conditions. The third strategy is the multi-stage recognition approach. This approach involves breaking the recognition process into two or more stages. In the early stages, a subword recognition is performed to obtain phonetic sequences that may or may not be in the recognizer's word vocabulary. By performing the recognition in two steps, the recognizer gets the chance to hypothesize novel phonetic sequences. The main drawback of the multi-stage approach is the fact the an important piece of knowledge, the word-level lexical knowledge, is not utilized early enough. Instead, it is delayed to some later stage causing performance on IV words to degrade. The fourth strategy is using filler models which is the most commonly used approach for handling OOV words. The approach involves adding a special lexical entry to the recognizer's vocabulary that represents an OOV word. A filler model typically acts as a generic word or a garbage model. The main drawback of filler models is the fact that they are highly unconstrained and can potentially absorb parts of the speech signal corresponding to in-vocabulary words.

Analysis of vocabulary growth and OOV words

In chapter 3, we looked at two characteristics of the vocabulary that influence the OOV problem: the growth of the vocabulary as a function of the size of the training corpus, and the coverage of that vocabulary of words on some unseen data. We found that for domain-dependent recognizers, vocabulary growth is much slower than for large vocabulary systems. The OOV rate varies from 1% to 3% depending on the size and selection of the vocabulary. Our findings were similar to those reported previously in this area. As to the nature of OOV words, we found that OOV words belong to one two categories: out-of-domain words, or domain-specific words that were left out because they were infrequent. At the phonetic-level we found that OOV words are on average the same length as IV words, but much longer when weighed by the frequency of usage. Finally, OOV words belong mainly to the three classes: nouns, verbs, and names.

A hybrid word-subword recognition framework

This thesis presented a novel approach for recognition of OOV words. A hybrid search space constructed by taking the union of a word-level and subword-level search spaces. The goal is to create two competing branches. The first branch is the IV branch that models the words

we know about, i.e. the words in the vocabulary. The second branch is the OOV branch that can generate phone sequences that are not in the baseline vocabulary. Even though our approach is similar to the idea of using filler models, the acoustic models typically used in filler models are much less detailed than those used by IV words. In our approach, when the OOV word is considered, the acoustic score is obtained from the same acoustic models used for IV words. These models are context-dependent and provide for a more accurate estimate of the acoustic score of the OOV word. Another important distinction for our approach is the use of an OOV language model to predict the presence of the such words in relation to other words in the utterance. Furthermore, the use of a statistical language model at the subword-level provides for more constraints during the search that can control what is being generated by the model, hence providing for more accurate subword recognition.

A dictionary-based approach for subword language modelling

The thesis also presented a dictionary-based approach for training subword language models that are used for recognition of OOV words. For the dictionary model, the OOV language model is trained from a large dictionary of words instead of a corpus of utterances. In this dictionary-based approach, n-grams are estimated from phone sequences of a large domain-independent word dictionary. This dictionary is significantly larger than the word vocabulary of the recognizer. By using a large vocabulary, we reduce domain-dependence bias; by training on vocabulary items, we avoid modelling cross-word phonotactics, and eliminate biasing the OOV network towards frequent words.

A mutual information approach for learning OOV multi-phone units

The thesis also explored an approach for learning multi-phone units for use in building a more accurate OOV model. The approach uses a bottom-up clustering algorithm that starts with the basic phone inventory and gradually merges pairs of phones to form longer multi-phone units. In deciding which pairs to merge into a new unit, the algorithm examines phone (or unit) co-occurrance statistics of all pairs occurring in a large domain-independent dictionary, and iteratively creates multi-phone units which are then used to create the OOV model. The criterion for merging a pair of units is based on the weighted mutual information of the pair. Large subword units provide for better constraints within the OOV search space where the only allowed phone sequences are the ones that can be formed by the multi-phone

units. With this automatic approach, we have control over the subword inventory size, and hence the size of the OOV model. When using a syllable inventory of words, we will need to include all syllables in the language in order to have complete coverage of all possible words.

An approach for generating novel phone sequences

We explored a variety of schemes to impose hard constraints on the OOV model. Such constraints are intended to completely prohibit certain phone sequences from being generated by the model. We explored constraints related to the length of OOV words including minimum and maximum length constraints. One important contribution of this thesis is the use of a complement model that guarantees that the OOV component generates only *novel* phone sequences. A novel phone sequence is one that does not match any phone sequence of any IV word. Generating a novel phone sequence guarantees that the hypothesized OOV word is a new word with a new phone sequence that does not belong to the vocabulary. To build such a complement model, IV words are represented by a finite state machine. This machine is then transformed using some basic finite state machine properties to accept only novel phone sequences.

A Multi-class extension for modelling OOV words

We also extended our framework to model multiple classes of OOV words. A multi-class model can better represent the contextual and linguistic relationships between the OOV word and its neighboring words, as well as to build more accurate OOV networks specialized to certain categories of words. Instead of augmenting the word search space with a single OOV model, we added several OOV models, one for each class of words. We explored two approaches for designing the OOV word classes. The first approach relies on using common part-of-speech tags. The second approach is a data-driven two-step clustering procedure, where the first step uses agglomerative clustering to derive an initial class assignment, while the second step uses iterative clustering to move words from one class to another in order to reduce the model perplexity.

Combining OOV modelling and confidence scoring

In chapter 8, we presented a method for combining OOV modelling with confidence scoring for the task of detecting speech recognition errors. Because of the inherently different nature of the two techniques, they have different advantages and disadvantages, and a combination of the two techniques proved beneficial. We presented a simple two-stage approach for combining the two techniques. The first stage involves *always* rejecting every OOV word hypothesis. The second stage examines the confidence scores of the remaining words in the hypothesized utterance. If the score is above some threshold, the word is accepted, otherwise it is rejected.

Experimental Findings

The thesis presented a series of experiments on a medium-vocabulary recognition task, JUPITER, and on a large-vocabulary task, HUB4. Experimental results demonstrated that for the medium-size vocabulary task the OOV approach is capable of detecting half of the OOV words for a very low false alarm rate. Detection can reach up to 70% for a false alarm rate less than 3%. The impact on IV performance is quite insignificant. IV WER suffers a small increase of when the model is introduced into a closed-vocabulary system. At this operating point, the word error rate (WER) on the IV utterances degrades slightly (from 10.9% to 11.2%) while the overall WER decreases from 17.1% to 16.4% due to correcting errors on words neighboring OOV words. Further improvement in WER is possible if the detected OOV words are identified in a second stage. The approach accurately locates most of the correctly detected OOV words. Most of the detected words fall within a window of 20 msec of the true boundary of the word. In addition the phonetic error rate on correctly detected OOV words is 31.2%. We also found that the approach works well in large vocabulary environments. The performance on HUB4 is slightly worse than that on JUPITER, and the overall WER is reduced by 1.4% (from 24.9% to 23.5%) at best.

We also presented a series of experiments to compare the mutual information model to the phone-based models. We found that the inventory of automatically learned units is mostly legal English syllables. We also presented detection and recognition results that showed that the mutual information model performs 11% better than the dictionary model and 30% better than a baseline corpus model in detecting OOV words. We also showed

that the PER improves by 18% over the dictionary OOV model.

Extending our approach to modelling multiple class of OOV words also proved beneficial. We presented a series of experiments to compare and contrast the part-of-speech and automatic approaches for acquiring OOV classes. We found that using the POS multi-class gives an FOM improvement of 6% over the single class model. The automatically-derived model improves FOM by 11%. The combination of the two approaches yields results which are only slightly better than the fully automatic approach with 13% FOM improvement over the single class model.

In experiments comparing OOV modelling and confidence scoring, we found that the OOV modelling approach does better at detecting OOV words while the confidence scoring approach performs better in detecting mis-recognitions in general. However, the combining the two approaches brings improvement of up to 25% in detecting recognition errors for domain keywords.

8.2 Future Work

While we have made much progress in tackling the OOV problem, there are many extensions that can done to either enhance performance or to make the approach applicable to a wider range of tasks related to spoken language systems. Next, we describe some of the possible future work.

Morphology-Based modelling of OOV words

In many situations new words are simply some variation on words we have in the vocabulary. These new words are usually formed by adding some suffix or prefix to the existing word or by compounding two existing words. Using morphology information, word structures can be learned and used to recognize new words. In systems like ANGIE [Seneff et al. 1996; Lau and Seneff 1997], a sub-lexical hierarchical model is used to represent word structure, the first layer of such structure is a morphology layer that captures word formation using a context-free grammar. A simple morphology-based approach can also be beneficial when considering some of the most common cases in the language. For example, explicitly modelling possible affixes for the language as part of the vocabulary and the grammar would allow for the recognition of OOV words formed by adding a prefix or a suffix to an in-vocabulary word.

Another approach is to automatically learn the syllable structure within the OOV network, without explicitly listing all possible syllables in the language.

Confidence scoring

There are numerous possible extensions to the work we presented on combining confidence scoring and OOV modelling. One extension is to develop confidence scoring methods specifically for determining whether a hypothesized OOV word is indeed OOV. Confidence features may include the subword n-gram score of the phonetic sequence, the difference between the acoustic score of the OOV word and the best IV word score, and the number of OOV words in the N-best list. A comparison of recognition paths containing and not containing a hypothesized OOV word could also provide suitable confidence measures for making this decision. This would allow us to build a confidence model specifically for OOV word detection. A second possible extension is to examine different methods for combining the two approaches. Running parallel recognizers and using a post-processing voting scheme is one possible alternative.

Modelling out-of-domain utterances

The idea of augmenting an IV search space with a subword search space to handle OOV words can also be applied at the whole utterance level for domain-dependent recognizers. The goal in this case will be to detect whether an utterance is an out-of-domain utterance. In systems such as the JUPITER weather system, users occasionally ask the system questions that are not related to its domain. In this case, the mismatch is not only at the vocabulary level but also at the language model level. An out-of-domain model can be constructed to contain an OOV component in addition to a domain-independent vocabulary and n-gram language model. Such an out-of-domain model will then be augmented with the in-domain search space. In this case, an input utterance will be fully hypothesized by either the in-domain recognition branch or the out-of-domain branch.

Combining the mutual information and multi-class approaches

Experimental results from Chapters 6 and 7 show that the mutual information model and the multi-class model both bring significant improvements in OOV detection. The mutual

information model expands the sub-lexical access within the OOV model to larger multi-phone units while the multi-class model expands OOV modelling into multiple classes while still using a phone-based sub-lexical access. Instead of using only phones, the multi-class model can use multi-phones units derived using the mutual information approach within each class of OOV words. After designing OOV classes and assigning class memberships, the mutual information approach can be applied at a class-specific level to learn the unit inventory and train the n-grams. Alternatively, the units can be learned in a class-independent fashion while the n-gram training can be class-specific.

Identifying OOV words

This thesis addressed the first three subproblems of the OOV problem: detecting the presence of OOV words, accurately locating them, and phonetically transcribing them. For certain applications such as a speech dictation system, providing a spelling of the word is important. The problem of identifying the OOV word is closely related to the sound-to-letter problem [Meng 1995; Hunnicutt et al. 1993] where given a *correct* phone sequence, the goal is to derive a spelling of the word. For a detected OOV word, the phonetic transcription may contain recognition errors, which makes the problem slightly different from the sound-to-letter problem. An approach for deriving a word spelling can benefit not only from the top best phonetic sequence generated by the subword recognizer but also from an N-best list of recognition hypothesis. For example if a large off-line dictionary exists, an approach can compare the N-best phone sequences to those in the dictionary to decide on the closest match in that dictionary.

Incorporating OOV words

Detecting an OOV word could be an indication that this word may be encountered by the recognizer in the future either by the same user or by different users. Hence, adding OOV words into the vocabulary can provide for better experience the next time around that word is presented to the system. Incorporating OOV words into the recognition systems requires updating the vocabulary list of the system as well as adding new n-grams into the language model to account for the newly added word in the context of in-vocabulary words.

Explicit modelling of partially-spoken words

A similar phenomenon to OOV words is that of partially spoken words, which are typically produced in more conversational or spontaneous speech applications. A user asking the system about the weather in a certain city may stop in the middle of uttering the city name and instead utters a different city name. This partial uttering of words also tend to produce errors since the recognizer matches the phonetic sequence with the best fitting words in its active vocabulary. Our approach can be specialized to handle the partial word phenomenon. One possibility is to create a pronunciation model of partially spoken words that can be based on some training data of partially spoken words. Alternatively, the approach can use complete word models but allow for exiting the model somewhere in the middle of the word but only at some syllable boundaries, the regions in the word where partial words may end.

Impact on the recognition search

One of the areas that we did not explore in this thesis is the impact of the OOV model on the recognition search. In [Hetherington 1994], it was shown that the presence of OOV words increases both the WER as well as the computational requirements of a closed-vocabulary recognizer. This increase in computation is due to the increased recognizer uncertainty and hence the increased number of word hypotheses competing with one another in the recognition search. With an explicit OOV model in place, this additional computational requirement still exists because the OOV model allows for a large number of pronunciations. One possible future work is to modify the search algorithm to allow for multiple pruning thresholds, one for the IV branch, another for the OOV branch. The main motivation for multiple thresholds is the presence of two n-gram language models of different size vocabularies (words and subwords), where n-gram probabilities have different dynamic ranges.

Appendix A

Initial Phone Inventory

The following table lists the phone label set used in the pronunciation dictionary for both the JUPITER recognizer as well as as in the large PRONLEX dictionary throughout this thesis. For each unit an example word is given with the portion of the word corresponding to the phone is emphasized.

Label	Example	Label	Example
aa	b*o*b	l	*l*ay
ae	b*a*t	m	*m*om
ah	b*u*t	n	*n*oon
ao	co*o*rdination	ng	si*ng*
aw	b*ou*t	nt	mai*nt*enance
ax	*a*bout	ow	b*oa*t
axr	count*er*plot	oy	b*oy*
ay	b*i*te	p	*p*ea
b	*b*ee	p-	s*p*ot
bd	a*b*duct (optional release)	pd	co*p*rocessor
ch	*ch*oke	r	*r*ay
d	*d*ay	s	*s*ea
dd	a*dd*s	sh	*sh*e
df	mu*dd*y	t	*t*ea
dh	*th*en	t-	s*t*op
dr	*dr*y	td	correc*t*ly
eh	b*e*t	tf	manslaugh*t*er
er	b*ir*d	th	*th*in
ey	b*ai*t	tq	bu*tt*on
f	*f*in	tr	*tr*y
g	*g*ay	uh	b*oo*k
gd	ne*g*ligence	uw	b*oo*t
hh	*h*ay	v	*v*an
ih	b*i*t	w	*w*ay
ix	deb*i*t	y	*y*acht
iy	b*ee*t	z	*z*one
jh	*j*oke	zh	mea*s*ure
k	*k*ey	-	utterance initial/final silence
k-	s*k*i	_	interword silence
kd	networ*k*s		

Table A-1: Label set used throughout this thesis. The first column shows the label, and the second gives an example word.

Appendix B

Pairs and Mutual Information Scores

The following two tables list the top 100 pairs for the mutual information approach presented in Chapter 5. The first table shows the top 100 pairs at the very beginning of the procedure. while the second table shows the top 100 pairs after 50 iterations. Only the top 10 pairs make it to the next iteration.

u_1	u_2	$10^3 MI_w$	u_1	u_2	$10^3 MI_w$	u_1	u_2	$10^3 MI_w$
ix	ng	83.581	ae	kd	7.737	w	ao	4.415
ax	n	63.778	p	r	7.529	n	dd	4.397
ax	l	29.005	r	ey	7.340	m	p	4.322
tr	r	28.997	tf	ix	7.324	m	ae	4.322
ao	r	28.078	ay	z	7.318	r	ih	4.267
r	iy	22.560	dr	r	7.271	k	ow	4.225
td	s	19.471	d	ax	7.251	ax	z	4.115
s	t-	18.313	ey	td	7.119	uh	r	4.102
tf	ax	18.166	tf	iy	7.012	nt	axr	4.085
sh	ax	17.585	r	ae	6.935	l	ey	4.027
y	uw	16.678	iy	z	6.881	jh	ax	4.017
axr	z	16.158	ax	m	6.556	f	ao	3.952
aa	r	15.414	s	t	6.359	d	iy	3.856
kd	s	15.112	n	td	6.153	y	ax	3.838
eh	kd	13.700	l	ax	6.090	nt	ax	3.700
ax	s	13.668	r	ow	5.883	m	aa	3.692
ey	sh	13.214	b	ax	5.599	s	ax	3.680
ae	n	12.828	ih	n	5.514	ax	tf	3.648
m	ax	12.132	g	r	5.459	b	er	3.621
l	iy	11.625	eh	r	5.453	ow	v	3.616
r	ax	11.559	w	ih	5.452	df	iy	3.597
s	k-	10.813	ah	n	5.407	f	ay	3.543
k	ax	10.708	v	axr	5.322	k	w	3.535
ey	tf	10.583	t	eh	5.201	ow	l	3.498
s	p-	10.406	ao	l	5.099	ax	k	3.433
s	td	10.361	r	eh	5.003	t	ey	3.414
eh	n	10.023	s	tr	4.968	v	ax	3.349
n	d	9.357	kd	t	4.811	d	ih	3.332
k	aa	9.132	k	ae	4.756	y	uh	3.321
n	z	9.054	ih	l	4.546	k	ao	3.303
df	ax	8.649	eh	l	4.545	m	ih	3.195
tq	en	8.339	ax	dd	4.522	hh	ae	3.145
aa	n	8.214	dd	z	4.468			
tf	axr	7.885	l	ay	4.432			

Table B-1: The top 100 pairs for the first iteration. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information.

u_1	u_2	$10^3 MI_w$	u_1	u_2	$10^3 MI_w$	u_1	u_2	$10^3 MI_w$
b	eh_l	1.862	ax	bd	1.672	zh	ax_n_z	1.573
w	ax	1.836	m	ax_kd	1.668	r_ax	b_ax_l	1.572
tr_r	ay	1.831	ax	b	1.663	f	y_uw	1.569
p_r	ay	1.817	d	ao_r	1.663	eh_kd_t	ax_v	1.562
t	er_n	1.794	r_ax	z	1.658	l	ae_n_dd	1.555
ae	r_ax	1.790	ax_tf_iy	z	1.653	tr_r	uw	1.553
k	ao_l	1.781	ao	l_ax	1.651	df	ax_m	1.547
d	ow	1.776	v	ax_n	1.651	k_aa	ng	1.544
eh_nt	ax_l	1.769	d	aa	1.651	ae	s_t	1.540
k	ao	1.764	tf_ax	kd	1.650	jh	ax_n	1.538
n	ey	1.762	ix_ng	z	1.646	r_ow	z	1.537
tf_ax	b_ax_l	1.754	ih	pd_s	1.646	hh	ih	1.527
l_ax_n	dd	1.751	ey_n	iy	1.645	ey_l	z	1.525
tf_iy	z	1.743	p	axr_z	1.640	gd	y_ax_l	1.523
n	axr	1.743	s_t-	aa_r	1.639	ax	r_ax	1.511
m_ay	kd	1.737	r_iy	n	1.632	ao	th	1.508
d	aw_n	1.730	eh_kd	td	1.615	df	ax_s	1.508
tr_r	ih	1.728	g	ae	1.613	l	dd	1.506
m	ih_l	1.727	g	aa	1.613	m_ae	gd_n	1.501
hh	iy	1.725	s_t-	ow_n	1.610	k	ax_s	1.500
b	oy	1.723	eh	td_s	1.609	ch	axr_z	1.499
p	ae_n	1.717	aw_n	dd	1.608	y_uh	r_ax	1.498
uh	kd	1.717	df	ax_k	1.605	t_ay	m	1.496
eh_kd	s_t-	1.716	aa	l	1.604	ax_r_iy	z	1.494
w	ih_l	1.710	ae_ng	g	1.602	d_ih	s_ax	1.493
ow_l	dd	1.708	d	ah	1.602	hh	y_uw	1.491
df	iy_z	1.707	ey_n	jh	1.602	th	er	1.486
aa_l_ax_jh	iy	1.705	t	aa	1.600	f	ey	1.479
y	ow	1.698	ae	gd	1.596	g	ah	1.478
ax_m	z	1.697	eh_r_iy	z	1.588	ih_kd	y_ax_l	1.474
pd	r_ax	1.691	jh_eh_n	ax_r	1.587	p	aa_n	1.471
k_r	uw	1.689	d	ax_m	1.586	s_k-	r	1.464
f_l	ae	1.677	f	ay_r	1.585			
bd	z	1.673	p	ow_z	1.579			

Table B-2: The top 100 pairs after iteration 50. The first column shows the first unit u_1, the second column shows the following unit u_2 and the third shows their weighted mutual information.

Appendix C

Utterance Level Confidence Features

One of the ten word-level features for confidence scoring in SUMMIT is the utterance level confidence score. For each utterance a single confidence feature is constructed from a set of utterance level features extracted from the recognizer. The following 15 utterance-level confidence features are the one used and are presented from [Hazen et al. 2000b]:

1. **Top-Choice Total Score:** The total score from all models (i.e., the acoustic, language, and pronunciation models) for the top-choice hypothesis.

2. **Top-Choice Average Score:** The average score per word from all models for the top-choice hypothesis.

3. **Top-Choice Total N-gram Score:** The total score of the N-gram model for the top-choice hypothesis.

4. **Top-Choice Average N-gram Score:** The average score per word of the N-gram model for the top-choice hypothesis.

5. **Top-Choice Total Acoustic Score:** The total acoustic score summed over all acoustic observations for the top-choice hypothesis.

6. **Top-Choice Average Acoustic Score:** The average acoustic score per acoustic observation for the top-choice hypothesis.

7. **Total Score Drop:** The drop in the total score between the top hypothesis and the second hypothesis in the N-best list.

8. **Acoustic Score Drop:** The drop in the total acoustic score between the top hypothesis and the second hypothesis in the N-best list.

9. **Lexical Score Drop:** The drop in the total N-gram score between the top hypothesis and the second hypothesis in the N-best list.

10. **Top-Choice Average N-best Purity:** The average N-best purity of all words in the top-choice hypothesis. The N-best purity for a hypothesized word is the fraction of N-best hypotheses in which that particular hypothesized word appears in the same location in the sentence.

11. **Top-Choice High N-best Purity:** The fraction of words in the top-choice hypothesis which have an N-best purity of greater than one half.

12. **Average N-best Purity:** The average N-best purity of all words in all of the N-best list hypothesis.

13. **High N-best Purity:** The percentage of words across all N-best list hypotheses which have an N-best purity of greater than one half.

14. **Number of N-best Hypotheses:** The number of sentence hypotheses in the N-best list. This number is usually its maximum value of ten but can be less if fewer than ten hypotheses are left after the search prunes away highly unlikely hypotheses.

15. **Top-Choice Number of Words:** The number of hypothesized words in the top-choice hypothesis.

Bibliography

Asadi, A., R. Schwartz, and J. Makhoul (1991). Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, pp. 305–308.

Asadi, A. O. (1991). *Automatic Detection and Modeling of New Words in a Large-Vocabulary Continuous Speech Recognition System.* Ph.D. thesis, Department of Electrical and Computer Engineering, Northeastern University, Boston.

Asadi, A. O. and H. C. Leung (1993). New-word addition and adaptation in a stochastic explicit-segment speech recognition system. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, pp. 642–645.

Bahl, L. R., F. Jelinek, and R. L. Mercer (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence PAMI 5*(2), 179–190.

Bazzi, I. and J. Glass (2000a). Heterogeneous lexical units for automatic speech recognition: Preliminary investigations. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Istanbul, pp. 1257–1260.

Bazzi, I. and J. Glass (2000b). Modelling out-of-vocabulary words for robust speech recognition. In *Proc. Int. Conf. Spoken Language Processing*, Beijing, pp. 401–404.

Bazzi, I. and J. Glass (2001). Learning units for domain-independent out-of-vocabulary word modelling. In *Proc. European Conf. Speech Communication and Technology*, Aalborg, pp. 61–64.

Bazzi, I. and J. Glass (2002). A multi-class approach for modelling out-of-vocabulary words. In *(submitted to) Int. Conf. Spoken Language Processing*, Denver.

Bellegarda, J. R., J. W. Butzberger, Y. Chow, N. B. Coccaro, and D. Naik (1996). A novel word clustering algorithm based on latent semantic analysis. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 172–175.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

Boros, M., M. Aretoulaki, F. Gallwitz, E. Noeth, and H. Niemann (1997). Semantic processing of out-of-vocabulary words in a spoken dialogue system. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1887–1890.

Brants, T. (1997). Internal and external tagsets in part-of-speech tagging. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 2787–2790.

Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Proc. Third Conference on Applied Natural Language Processing*, Trento, Italy. Association of Computational Linguistics.

Brill, E. (1993). *A Corpus-Based Approach to Language Learning.* Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

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

Bub, T. and J. Schwinn (1996). Verbmobil: The evolution of a complex large speech-to-speech translation system. In *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, pp. 2371–2374.

Bub, T., W. Wahlster, and A. Waibel (1997). Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 71–74.

Chang, J. W. (1998). *Near-Miss Modeling: A Segment-Based Approach to Speech Recognition.* Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge.

Chang, J. W. and J. R. Glass (1997). Segmentation and modeling in segment-based recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1199–1202.

Chase, L. (1997a). Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 815–818.

Chase, L. (1997b). Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 815–818.

Chen, S. F. and J. Goodman (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

Chow, Y. L., M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos, and R. M. Schwartz (1990). BYBLOS: The BBN continuous speech recognition system. In A. Waibel and K.-F. Lee (Eds.), *Readings in Speech Recognition*, pp. 596–599. San Mateo, CA: Morgan Kaufmann.

Chung, G. (2001). *Towards Multi-Domain Speech Understanding with Flexible and Dynamic Vocabulary.* Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

De Mori, R. and M. Galler (1996). The use of syllable phonotactics for word hypothesization. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 877–880.

Deligne, S. and F. Bimbot (1995). Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Detroit, pp. 169–172.

Deligne, S. and F. Bimbot (1997). Inference of variable-length acoustic units for continuous speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 1731–1734.

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

Deng, L. and H. Sameti (1994). Automatic speech recognition using dynamically defined speech units. In *Proc. Int. Conf. Spoken Language Processing*, Yokohama, pp. 2167–2170.

Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis.* New York, NY: John Wiley & Sons.

Farhat, A., J. Isabelle, and D. O'Shaughnessy (1996). Clustering words for statistical language models based on contextual word similarity. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 180–183.

Fetter, P., A. Kaltenmeier, T. Kuhn, and P. Regel-Brietzmann (1996). Improved modeling of oov words in spontaneous speech. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 534–537.

Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara.

Gallwitz, F., E. Noeth, and H. Niemann (1996). A category based approach for recognition of out-of-vocabulary words. In *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, pp. 228–231.

Geutner, P. (1997). Fuzzy class rescoring: A part-of-speech language model. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 2743–2746.

Gillick, L., Y. Ito, and J. Young (1997). A probabilistic approach to confidence estimation and evaluation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 879–882.

Glass, J. (1988). *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition.* Ph. D. thesis, Massachusetts Institute of Technology, Cambridge.

Glass, J., J. Chang, and M. McCandless (1996). A probabilistic framework for feature-based speech recognition. In *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, pp. 2277–2280.

Glass, J. R., Hazen, T. J., and I. L. Hetherington (1999). Real-time telephone-based speech recognition in the JUPITER domain. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Phoenix, pp. 61–64.

Glass, J. R. and T. J. Hazen (1998). Telephone-based conversational speech recognition in the JUPITER domain. In *Proc. Int. Conf. Spoken Language Processing*, Sydney, pp. 1327–1330.

Godfrey, J., E. Holliman, and J. McDaniel (1992). SWITCHBOARD: telephone speech corpus for research and development. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco.

Gorin, A. L., G. Riccardi, and J. H. Wright (1997). How may I help you? *Speech Communication 23*(1/2), 113–127.

Graff, D., W. Z. M. R. and M. Liberman (1997). The 1996 broadcast news speech and language-model corpus. In *Proc. DARPA Speech Recognition Workshop '97*, Chantilly.

Grishman, R., C. Macleod, and S. Wolff (1994). The comlex syntax project. In *Proc. ARPA Human Language Technology Workshop '93*, Princeton, NJ, pp. 300–302. distributed as *Human Language Technology* by San Mateo, CA: Morgan Kaufmann Publishers.

Halberstadt, A. K. (1998). *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition.* Ph. D. thesis, Massachusetts Institute of Technology, Cambridge.

Hayamizu, S., K. Itou, and K. Tanaka (1993). Detection of unknown words in large vocabulary speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Berlin, pp. 2113–2116.

Hazen, T. and I. Bazzi (2001). A comparison and combination of methods for oov word detection and word confidence scoring. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Salt Lake City, pp. 397–400.

Hazen, T. J., T. Burianek, J. Polifroni, and S. Seneff (2000a). Integrating recognition confidence scoring with langauge understanding and dialogue modeling. In *Proc. Int. Conf. Spoken Language Processing*, Beijing.

Hazen, T. J., T. Burianek, J. Polifroni, and S. Seneff (2000b). Recognition confidence scoring for use in speech understanding systems. In *Proceedings 2000 IEEE Workshop on Automatic Speech Recognition and Understanding*, Paris, France.

Hetherington, I. L. (1994). *The Problem of New, Out-of-Vocabulary Words in Spoken Language Systems.* Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Hetherington, I. L. (2001). An efficient implementation of phonological rules using finite-state transducers. In *Proc. European Conf. Speech Communication and Technology*, Aalborg.

Hu, Z., J. Schalkwyk, E. Barnard, and R. A. Cole (1996). Speech recognition using syllable-like units. In *Proc. Int. Conf. Spoken Language Processing*, Volume 2, Philadelphia, pp. 1117–1120.

Hunnicutt, S., H. Meng, S. Seneff, and V. Zue (1993). Reversible letter-to-sound sound-to-letter generation based on parsing word morphology. In *Proc. European Conf. Speech Communication and Technology*, Berlin, pp. 763–766.

Itou, K., S. Hayamizu, and H. Tanaka (1992). Detection of unknown words and automatic estimation of their transcriptions in continuous speech recognition. In *Proc. Int. Conf. Spoken Language Processing*, Banff, pp. 799–802.

Jardino, M. (1996). Multilingual stochastic n-gram class language models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 161–164.

Jelinek, F., R. Mercer, and S. Roukous (1990). Classifying words for improved statistical language models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, pp. 621–624.

Jones, M. and P. C. Woodland (1994). Modelling syllable characteristics to improve a large vocabulary continuous speech recogniser. In *Proc. Int. Conf. Spoken Language Processing*, Yokohama, pp. 2171–2174.

Jones, R. J. (1996). *Syllable-based word recognition.* Ph. D. thesis, Department of Electrical and Electronic Engineering, Univeristy of Wales Swansea, Wales, U.K.

Jones, R. J., S. Downey, and J. S. Mason (1997). Continuous speech recognition using syllables. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1171–1174.

Jusek, A., G. Fink, H. Rautenstrauch, and Sagerer (1995). Detection of unknown words and its evaluation. In *European Conf. Speech Communication and Technology*, Madrid, pp. 2107–2110.

Kamppari, S. O. and T. J. Hazen (2000). Word and phone level acoustic confidence scoring. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Istanbul, Turkey, pp. 1799–1802.

Kemp, T. and A. Jusek (1996). Modelling unknown words in spontaneous speech. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 530–533.

Kita, K., T. Ehara, and T. Morimoto (1991). Processing unknown words in continuous speech recognition. *IEICE Trans. E74*(7), 1811–1816.

Klakow, D., G. Rose, and X. Aubert (1999). OOV detection in large vocabulary system using automatically defined word-fragments as fillers: Detection of unknown words and its evaluation. In *European Conf. Speech Communication and Technology*, Budapest, pp. 49–52.

Kneser, R. and J. Peters (1997). Semantic clustering for adaptive language modeling. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 779–782.

Kubala, F., H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, and J. Makhoul (1997). Advances in transcription of broadcast news. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 927–930.

Lau, R. and S. Seneff (1997). Providing sublexical constraints for word spotting within the ANGIE framework. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 263–266.

Lee, S. C. (1998). *Probabilistic Segmentation for Segment-Based Speech Recognition.* M. S. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge.

Livescu, K. (1999). Analysis and modeling of non-native speech for automatic speech recognition. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Manos, A. S. (1996). A study on out-of-vocabulary word modelling for a segment-based keyword spotting system. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Manos, A. S. and V. W. Zue (1997). A segment-based wordspotter using phonetic filler models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 899–902.

McCandless, M. and J. Glass (1994). Empirical acquisition of language models for speech recognition. In *Proc. Int. Conf. Spoken Language Processing*, Yokohama.

McCandless, M. K. (1994). Automatic acquisition of language models for speech recognition. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

McLemore, C. (1997). PRONLEX American English Lexicon. URL http://morph.ldc.upenn.edu/Catalog/LDC97L20.html.

Meng, H. M. (1995). *Phonological Parsing for Bi-directional Letter-to-Sound / Sound-to-Letter Generation.* Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Meng, H. M., S. Seneff, and V. W. Zue (1994). Phonological parsing for reversible letter-to-sound / sound-to-letter generation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, pp. II–1 – II–4.

Mohri, M. and M. Riley (1997). Weighted determinization and minimization for large vocabulary speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 131–134.

Ng, K. and V. Zue (1997). An investigation of subword unit representations for spoken document retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 339.

Niesler, T. and P. Woodland (1996). A variable-length category-based n-gram language model. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 164–167.

Pao, C., P. Schimdt, and J. Glass (1998). Confidence scoring for speech understanding system. In *Proc. Int. Conf. Spoken Language Processing*, Sydney.

Paul, D. and J. Baker (1992). The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, NY, pp. 357–362.

Pfau, T., M. Beham, W. Reichl, and G. Ruske (1997). Creating large subword units for speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1191–1194.

Polifroni, J., S. Seneff, and V. W. Zue (1991). Collection of spontaneous speech for the ATIS domain and comparative analyses of data collected at MIT and TI. In *Proceedings of the DARPA Speech and Natural Language Workshop.*

Price, P., W. Fisher, J. Bernstein, and D. Pallett (1988). The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proceedings of the 1988 International Conference on Acoustics, Speech and Signal Processing*, New York, NY, pp. 651–654.

Rabiner, L. and B. Juang (1993). *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice-Hall.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE 77*, 257–286.

Rivlin, Z., M. Cohen, V. Abrash, and T. Chung (1996). A phone-dependent confidence measure for utterance rejection. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 515–518.

Schaaf, T. and T. Kemp (1997). Confidence measures for spontaneous speech. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 875–878.

Seneff, S. (1992). TINA: A natural language system for spoken language applications. *Computational Linguistics 18*(1), 61–86.

Seneff, S., R. Lau, and H. Meng (1996). ANGIE: A new framework for speech analysis based on morpho-phonological modelling. In *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, pp. 110–113.

Sipser, M. (1997). *Introduction to the Theory of Computation.* Cambridge, MA: PWS Publishing Company.

Siu, M.-H., H. Gish, and F. Richardson (1997). Improved estimation,evaluation and applications of confidence measures for speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 831–834.

Spina, M. S. and V. Zue (1997). Automatic transcription of general audio data: Effect of environment segmentation on phonetic recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1547–1550.

Suhm, B., M. Woszczyna, and A. Waibel (1993). Detection and transcription of new words. In *Proc. European Conf. Speech Communication and Technology*, Berlin, pp. 2179–2182.

Tamoto, M. and T. Kawabata (1995). Clustering word category based on binomial posteriori co-occurence distribution. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Detroit, pp. 165–168.

Ward, W. and S. Issar (1996). A class based language model for speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, pp. 416–419.

Weintraub, M., F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke (1997). Neural-network based measures of confidence for word recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 887–890.

Williams, G. and S. Renals (1997). Confidence measures for hybrid hmm/ann speech recognition. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 1955–1958.

Winston, P. (1992). *Artificial Intelligence* (Third ed.). Reading, MA: Addison-Wesley.

Woodland, P. C., M. J. Gales, D. Pye, and S. J. Young (1997). Broadcast news transcription using htk. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, pp. 719–722.

Zue, V., J. Glass, D. Goodine, M. Phillips, and S. Seneff (1990). The SUMMIT speech recognition system: Phonological modelling and lexical access. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, pp. 49–52.

Zue, V., S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington (2000). JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing 8*(1), 100–112.

Zue, V., S. Seneff, J. R. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid (1997). From interface to content: Translingual access and delivery of on-line information. In *Proc. European Conf. Speech Communication and Technology*, Rhodes, pp. 2227–2230.