Linguistically-Motivated Sub-word Modeling with Applications to Speech Recognition

by

Ghinwa F. Choueiter

B.E., American University of Beirut, Lebanon (2002)
S.M., Massachusetts Institute of Technology, Cambridge, MA (2004)

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

February 2009

Author ....
Department of Electrical Engineering and Computer Science
January 9, 2009

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

Certified by..
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Linguistically-Motivated Sub-word Modeling with Applications to Speech Recognition

by

Ghinwa F. Choueiter

Submitted to the Department of Electrical Engineering and Computer Science
on January 9, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Despite the proliferation of speech-enabled applications and devices, speech-driven human-machine interaction still faces several challenges. One of theses issues is the new word or the out-of-vocabulary (OOV) problem, which occurs when the underlying automatic speech recognizer (ASR) encounters a word it does not "know". With ASR being deployed in constantly evolving domains such as restaurant ratings, or music querying, as well as on handheld devices, the new word problem continues to arise.

This thesis is concerned with the OOV problem, and in particular with the process of modeling and learning the lexical properties of an OOV word through a linguistically-motivated sub-syllabic model. The linguistic model is designed using a context-free grammar which describes the sub-syllabic structure of English words, and encapsulates phonotactic and phonological constraints. The context-free grammar is supported by a probability model, which captures the statistics of the parses generated by the grammar and encodes spatio-temporal context. The two main outcomes of the grammar design are: (1) sub-word units, which encode pronunciation information, and can be viewed as clusters of phonemes; and (2) a high-quality alignment between graphemic and sub-word units, which results in hybrid entities denoted as spellnemes. The spellneme units are used in the design of a statistical bi-directional letter-to-sound (L2S) model, which plays a significant role in automatically learning the spelling and pronunciation of a new word.

The sub-word units and the L2S model are assessed on the task of automatic lexicon generation. In a first set of experiments, knowledge of the spelling of the lexicon is assumed. It is shown that the phonemic pronunciations associated with the lexicon can be successfully learned using the L2S model as well as a sub-word recognizer. In a second set of experiments, the assumption of perfect spelling knowledge is relaxed, and an iterative and unsupervised algorithm, denoted as Turbo-style, makes use of spoken instances of both spellings and words to learn the lexical entries in a dictionary.

Sub-word speech recognition is also embedded in a parallel fashion as a backoff mechanism for a word recognizer. The resulting hybrid model is evaluated in a lexical access application, whereby a word recognizer first attempts to recognize an isolated word. Upon failure of the word recognizer, the sub-word recognizer is manually triggered. Preliminary results show that such a hybrid set-up outperforms a large-vocabulary recognizer.

Finally, the sub-word units are embedded in a flat hybrid OOV model for continuous ASR. The hybrid ASR is deployed as a front-end to a song retrieval application, which is queried via spoken lyrics. Vocabulary compression and open-ended query recognition are achieved by designing a hybrid ASR. The performance of the front-end recognition system is reported in terms of sentence, word, and sub-word error rates. The hybrid ASR is shown to outperform a word-only system over a range of out-of-vocabulary rates (1%-50%). The retrieval performance is thoroughly assessed as a function of ASR *N*-best size, language model order, and the index size. Moreover, it is shown that the sub-words outperform alternative linguistically-motivated sub-lexical units such as phonemes. Finally, it is observed that a dramatic vocabulary compression - by more than a factor of 10 - is accompanied by a minor loss in song retrieval performance.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Acknowledgments

Meme s'il a perdu une plume ce canard, il a encore un million de plumes!

Raphaella Khoury

I would first like to extend my deepest gratitude to my thesis advisors, Jim Glass and Stephanie Seneff. Jim's endless patience and Stephanie's vivacious personality provided the perfect balance for me to complete my research and my thesis. I am thankful to Jim for taking me on as a student at the Spoken Language Systems (SLS) in the summer of 2003. Since then, he has helped me grow into the speech researcher I am today. Stephanie introduced me to my doctoral thesis topic, and her constant enthusiasm provided me with gentle nudges towards the finish line.

I would also like to thank the members of my thesis committee, Prof. Louis Braida and Prof. Michael Collins. I am grateful for their helpful suggestions and comments which guided me whenever I lost sight of the big picture. Prof. Braida's broad questions helped me keep the lay audience in mind, and Prof. Collins's technical questions helped me think about the problem and the approach more rigorously.

I would not have gone to MIT, nor would I have reached this point in my life, had it not been for Mesrob Ohannessian. I have been blessed enough to have him as a true friend who loves me unconditionally and shares my joys and pains every step of the way. Looking back, it seems that there has not been a single topic about life or research that I have not discussed in depth with him. Side Note - Mesrob is also a collaborator on the research presented in Chapter 5.

As a graduate student at MIT, I had the opportunity to intern in several research institutions where I learned more about different aspects of speech technologies. I am grateful to Geoffrey Zweig who first took me on as an intern at IBM, T.J. Watson Research Center in the summer of 2005. While at IBM, I was lucky enough to work with Dan Povey and Stanley Chen on an Arabic recognition project. Geoffrey Zweig also gave me the chance to intern at Microsoft Research in the summer of 2007, where I collaborated with him and Patrick Nguyen on an accent classification project. I aspire to, someday, acquire the excellent research skills that I found in Geoffrey, Dan, Stanley, and Patrick. I spent the summer of 2006 at the American University of Beirut (AUB) in my country, Lebanon. I had been invited by Prof. Al-Alaoui to work as a research associate on a computer-aided Arabic tutoring tool. Prof. Al-Alaoui, who was also my advisor while I was an undergraduate at AUB, introduced me to the world of research, and will always be a valuable mentor.

Warm thanks go to all the members of the SLS group, staff and students, for the opportunity to have them as colleagues and friends. A special thank you goes to Marcia Davidson, the administrative assistant of the group, who always cheered me up with her priceless witty remarks [1]. I am grateful to the past and current SLS

[1] Advice to new and future students: Keep Marcia happy!.

students, John Lee, Ken Schutte, Mitch Peabody, Alex Park, Ed Filisko, Ian McGraw, Alex Gruenstein, Tara Sainath, and Hung-an Chang. My stay at SLS would not have been quite the same without them. Thank you to SLS researchers, Scott Cyphers and Lee Hetherington, who answered my many questions on computer and software maintenance [2]. I would also like to thank Lee and past SLS researchers, T.J. Hazen and Chao Wang, who taught me most of what I know about the speech recognizer infrastructure in the laboratory. Special thanks go to Stanley Wang and Ibrahim Badr, who helped set up the experiments presented in Chapter 7.

To Ruaidhri O'Connor who taught me about human strengths but more importantly about human weaknesses.

To Ari Shapiro who sows smiles and orderly chaos wherever he goes. I cannot thank him enough. Thank you to Jim Geraci, who always knows how to wind me up, Michael Bernstein, whose enthusiasm is contagious and whose nabaztag should be destroyed(!), and to Katrina Panovich, who reminds me of my young self back in the day.

Throughout my stay at MIT and in Cambridge, I have had the chance to become friends with individuals who left a significant mark in my life. My space-mate Gregory Marton (Gremio) at CSAIL is one of those ”markers” whose conversations about life and work I will treasure. I look forward to many more conversations to come. As a member of the Lebanese Student Club @ MIT, I looked up to - and still do - Loai Naamani, Nader Shaar, Rabih Zbib, and Fadi Kanaan. Their perseverance and dedication, no matter the task at hand, give a new meaning to the term ”professionalism”. I am thankful to have them as friends and look forward to future ventures with them. SLS alumni, Karen Livescu and Han Shu, gave me hope that there is light at the end of the tunnel. Karen and Han endured all my newbie questions when I first joined SLS, and always lent me an ear no matter what I needed to talk about. As a choir member at the Jesuit Urban Center, I got to meet Ellen Oak, Cindy O'Meara, Linda Teuwen, Mark Brown, Dong-ill Shin, Peter Wick, Kira Hanson, etc. They always managed to brighten my thursdays and sundays and filled them with song and music. During my first year at MIT, I lived in the Green Hall dorm, where I was fortunate enough to meet Rayka Yokoo, Mana Taghdiri, and Roya Beheshti. Though nothing replaces family, their friendship made it easier to be away from home.

To my parents Fakhry and Therese Choueiter, to my sisters Nadine and Mary. I love you. I cannot put into words how grateful I am for all your love and endless sacrifices. I am thankful for every phone call - even the ones at six in the morning - and every visit - even when I was close to a deadline. I hope to someday acquire even a small portion of my mother's perseverance and my father's dedication to his work and family. I am grateful for my sisters who never failed to be there for me through laughs and tears. I could not have done it without you. I DID IT!!

This research was supported by the Industrial Technology Research Institute in Taiwan and by Nokia, as part of a joint MIT-Nokia research collaboration.

[2] To Scott, Lee: it really wasn't me!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Despite the significant improvements achieved in automatic speech recognition (ASR) systems over the last several decades [Glass, 2003; Lamere et al., 2003; Prasad et al., 2005; Chen et al., 2006], and the proliferation of speech-enabled applications and devices [Chang et al., 2002; Gorin et al., 1997; Muthusamy et al., 1999; Zue et al., 2000; Vlingo], speech-driven human-machine interaction still faces several challenges. One of theses issues is the new word or the out-of-vocabulary (OOV) problem, which occurs whenever the underlying ASR encounters a word that it does not "know".

This thesis is concerned with the OOV problem and in particular with the process of modeling and learning the lexical properties of an OOV word. To appreciate the new word issue, it is important to understand what it means for an ASR system to *know* a word. ASR is the process of decoding a spoken utterance into a string of words. The prevailing approach to ASR is to model the spoken utterance as a weighted network of sub-lexical units, which are typically phones, the smallest distinguishable speech sounds in a language. A phone graph models all possible speech sounds that correspond to the input acoustic signal, and is constrained by three knowledge sources: (1) **the acoustic model**, which statistically models context-dependent or context-independent phones, and is trained on the acoustic-phonetic measurements extracted from the speech signal, (2) **the lexicon**, also known as a phonetic dictionary, which typically maps words to their phonetic pronunciations, and (3) **the language model**, which models the probability of a word sequence. A search through the constrained phone graph gives a string of words that best matches the input acoustic signal. Ideally, for an ASR system to know a word, the acoustic model should be able to appropriately model the phonetic representation of the word, the lexicon should contain the word and its pronunciation, and the language model should reliably predict the occurrence of the word [Hetherington, 1995]. Figure 1-1 illustrates the aforementioned ASR process, and demonstrates a potential ASR outcome when the system is presented with a word that it does not know, in this case *euthanasia*. Since the ASR system is not designed to deal with new words, it hypothesized word sequences that closely matched the input acoustically.

The new word problem is an important one, and with ASR being deployed in constantly evolving domains such as broadcast news transcription, restaurant rating, or music querying, such an issue continues to arise. The root of the problem lies in

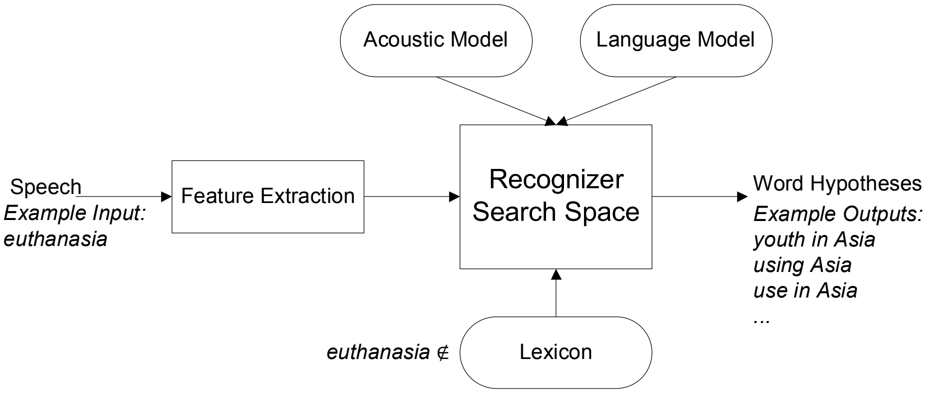


Figure 1-1: A block diagram of a standard speech recognition system, which decodes an acoustic signal into a string of words guided by an acoustic model, a lexicon, and a language model. The figure illustrates the erroneous recognition outputs when the ASR system is presented with a word, *euthanasia*, that is not in its lexicon.

having the basic lexical unit in ASR be a word. Hence, if a word is not in the ASR lexicon, there is no way of hypothesizing it, and the system will always produce an error. Table 1.1 illustrates this idea through a dialogue snippet from a flight reservation domain [Seneff, 2002], where the user wishes to reserve a flight to *Dominica*, a destination unknown to the ASR system. If the dialogue system had some mechanism to detect the occurrence of new words and learn them, then the human-machine interaction would become less frustrating and more useful to the user. Whereas humans can naturally recognize new nonsense syllables and words [Lippmann, 1997], ASR systems with static lexicons and simple lexical access techniques lag in their ability to detect a new word, let alone learning and adding it to the lexicon. As will be shown in Section 1.1, despite the research invested in the new-word problem, there remains much room for improvement.

With this in mind, the focus of this thesis is two-fold:

- **Automatic lexical learning and generation**, which addresses the lexical aspect of acquiring a new word, and which involves learning the graphemic orthography and the phonetic representation of a new word.

- **Sub-word modeling and recognition**, which addresses the more general problem of OOV modeling in ASR, and which involves modeling words using sub-word units.

In this thesis, we propose to use linguistically-motivated sub-syllabic sub-word units as the basic recognition units instead of whole words. The sub-word units are intermediate forms between phonemes and syllables, and are designed to model positional and phonological constraints while encoding pronunciation information. One can view the sub-syllabic sub-word units as clusters of one or more phonemes. In the process of designing the sub-word units, a set of hybrid entities denoted as

U	I would like to fly from Miami, Florida to the island of Dominica [*Seattle and Atlanta*] on December 17th.
S	Okay, from Miami to Seattle on Monday December 17th. Can you provide a departure time or airline preference?
U	No, I don't want to go to Seattle.
S	What city does the flight arrive in?
U	Roseau, Dominica [*Toronto, Connecticut*].
S	Okay, from Miami to Toronto on Monday December 17th. Can you provide a departure time or airline preference?
U	No, I'm not going to Toronto. I want to go to Dominica [*Jamaica*].
S	Okay, from Miami to Kingston on Monday December 17th. Can you provide a departure time or airline preference?
U	No, I'm not going to Kingston [*Houston*], I want to go to Dominica [*Connecticut*].

Table 1.1: Sample dialogue from the Mercury flight reservation domain [Seneff, 2002] where the user, U, wishes to reserve a flight to a city that the system, S, does not know. [*Italic words*] correspond to what the system actually recognized. (From [Filisko and Seneff, 2005])

spellnemes[1] are developed. Whereas the sub-word units encode only pronunciation, the spellnemes encode both spelling and pronunciation. Given the hybrid nature of the spellnemes, they can be used, in a fairly straightforward manner, to design bi-directional letter-to-sound/sound-to-letter (L2S/S2L) models. At this point, we note the following remarks:

- Automatic lexical learning and generation can be achieved with the help of bi-directional L2S/S2L conversion. For example, if a trained L2S model is presented with a letter sequence, it could generate its corresponding phonetic (sound) representation, and a lexical entry would be produced.

- When sub-word units are incorporated into an ASR, the recognizer becomes open-ended and can potentially model any word, including OOV ones, as a sequence of sub-word units.

The aforementioned points constitute the core of the research presented in this thesis, and will be covered in more depth in later chapters.

[1] The term spellneme stands for spelling and phoneme, where the phoneme is the smallest abstract unit in the sound system of a language that distinguishes meaning. In this research, a spellneme is one or more letters concatenated with one or more phonemes. A spellneme is also denoted as a graphone in the literature (grapheme and phoneme) [Bisani and Ney, 2005].

In the rest of this thesis, we refer to the linguistically-motivated sub-syllabic units as simply sub-word units unless clarification is required, in which case we refer to them by their full name.

In the rest of this chapter, we further motivate the new word problem and the proposed approach through a literature review of OOV modeling, sub-word modeling, and L2S/S2L conversion. We then describe the approach, the thesis contributions, as well as the outline of the remaining chapters.

1.1 Motivation

1.1.1 The New Word Problem

The rate of new word occurrence is tied to the design of the ASR vocabulary. Previous studies have shown that it is practically impossible to design a vocabulary capable of covering all possible speech input [Hetherington, 1995]. Moreover, constantly increasing the vocabulary size is bound to introduce acoustic ambiguity and worsen ASR performance [Rosenfeld, 1995]. Hence, to address the new word problem, ASR should undergo a paradigm shift from vocabulary design to that of an adaptive system that can reliably detect and learn new words.

A thorough study of the new word problem was presented in [Hetherington, 1995], where nine corpora covering multiple languages and applications were examined. The applications consisted of human-computer interactive problem solving with small vocabularies, spontaneous human-human interaction with medium vocabularies, and newspaper text with large vocabularies. Sample corpora that corresponded to the aforementioned applications were Voyager [Zue et al., 1989a], Switchboard [Godfrey et al., 1992], and the Wall Street Journal (WSJ) [Paul and Baker, 1992]. The languages covered were English, French, and Italian. In Figure 1-2 from [Hetherington, 1995], the vocabulary growth is illustrated for each of the nine studied corpora. The largest vocabulary growths correspond to the news transcription corpora, which are relatively open-ended. The smallest vocabulary growths correspond to the limited-domain human-computer interactive problem solving corpora. More importantly, even for a large number of training words such as the WSJ or Switchboard, the vocabulary growth does not plateau.

Figure 1-3, also from [Hetherington, 1995], illustrates the rate of new words as a function of number of training words over the nine corpora. The results in Figure 1-3 are consistent with those in Figure 1-2 indicating that corpora with large vocabulary growth also exhibit high new word rates, more commonly known as OOV rates[2]. Moreover, although the OOV word rate decreases with training data size, it never reaches zero. It is further shown in [Hetherington, 1995] that it can take a vocabulary size at least as large as 100,000 words to reduce the OOV rate below 1%. Although a 1% OOV rate might seem to have little impact on ASR performance, it could correspond to a 17% OOV utterance rate, i.e. 17% of the utterances have at least

[2] *OOV rate* typically refers to OOV word rate, i.e. the rate at which OOV words occur in a particular text. A detailed description of the OOV rate is provided in Section 7.3.1.

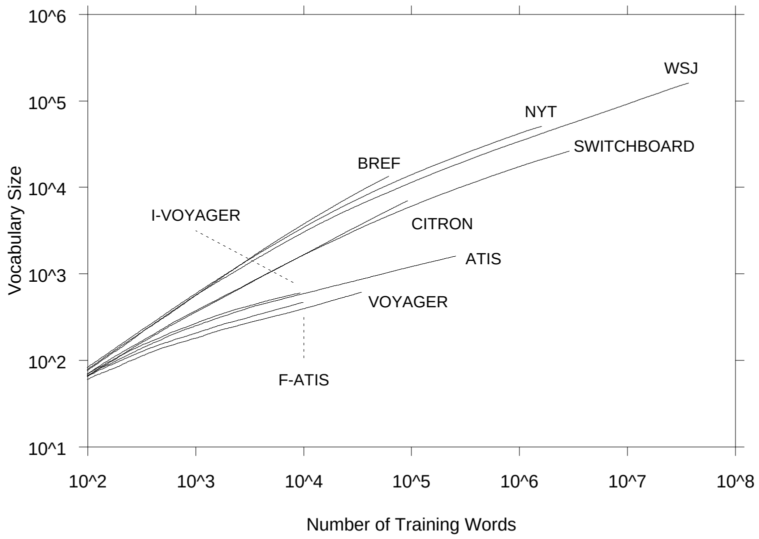


Figure 1-2: The growth of the vocabulary size as a function of the number of training words for nine corpora spanning the English, French, and Italian languages. (From [Hetherington, 1995])

one OOV word. This could have adverse effects on a user's interaction with a spoken dialogue system.

In [Fang and Huckvale, 2000], different-sized vocabularies were drawn from 90% of the 100-million-word British National Corpus (BNC) [Burnard, 1995], and were evaluated in terms of OOV rates on several English corpora. The test corpora included the remaining 10% of the BNC, the International Corpus of English (ICE) [Greenbaum, 1996], the Survey of English Usage (SEU) [Greenbaum and Svartvik, 1990], and the Lancaster-Oslo-Bergen Corpus of English (LOB) [Hofland and Johansson, 1982]. The first three test corpora consisted of transcribed speech and written text, and were split accordingly, e.g. ICE-S (speech) and ICE-W (written). LOB consisted only of written text. Table 1.2 illustrates the OOV rates on the test corpora for the different vocabulary sizes. The results were consistent with those reported in [Hetherington, 1995], whereby test data corresponding to written text exhibited higher OOV rates than those corresponding to transcribed speech. Moreover, the lowest OOV rate was 0.7% which is still significant. It was also shown that, as the 100-million-word BNC was swept, the vocabulary size grew to ∼600k unique words with no indication of a plateau.

In [Bazzi, 2002], OOV modeling was introduced to Jupiter, a weather domain dialogue system [Zue et al., 2000]. An analysis of the effect of OOV words on Jupiter utterances was also conducted and sample results were reported in Figure 1-4. The results were reported for in-vocabulary (IV) and out-of-vocabulary utterances separately in terms of word error rate (WER) and sentence error rate (SER). Figure 1-4 illustrates a major consequence of OOV words: *the ripple effect*, whereby not only are OOV words misrecognized, but potentially, so are the neighboring words. This explains the fact that the WER of OOV utterances is nearly four times that of IV

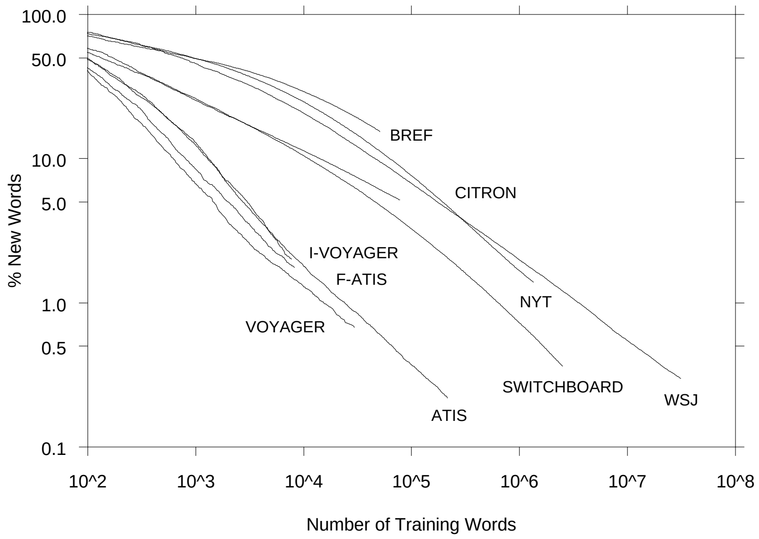


Figure 1-3: The new word rate as a function of the number of training words for nine corpora. Though the new word rate decreases with the size of the training data, it never reaches zero. (From [Hetherington, 1995])

utterances.

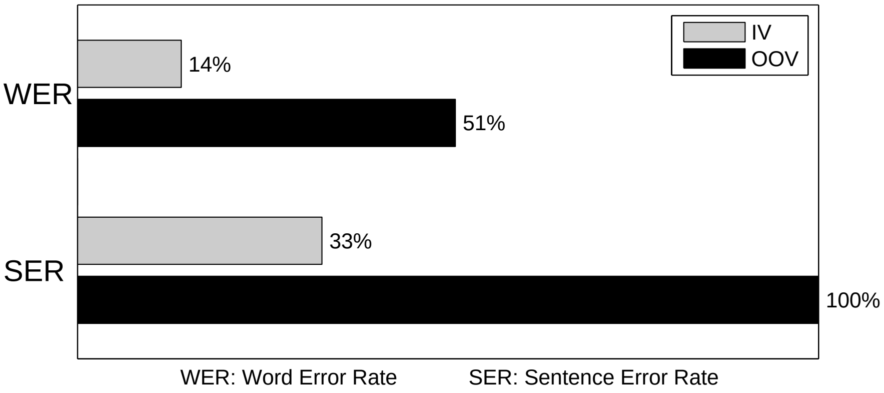


Figure 1-4: The WER and SER for in-vocabulary and out-of-vocabulary utterances from the JUPITER domain. The WER of OOV utterances is nearly four times worse than that of IV utterances. (From [Bazzi, 2002])

More recently, it has been shown that new words are responsible for performance breakdown even in the most state-of-the-art ASR systems [Furui et al., 2005; Duta et al., 2006]. An analysis of the errors produced by the BBN RT04 (Rich Text) ASR system [Nguyen et al., 2005; Prasad et al., 2005] in the DARPA EARS evaluations [Wayne, 2003] showed that 10-15% of the errors on broadcast news occurred due to named entities which are mostly poorly trained or OOV words [3] [Duta et al., 2006].

[3] OOV rates on the broadcast news test sets were quite low and ranged between 0.2% and 0.7%

Corpus	20k	40k	60k	80k	100k
BNC	3.6	1.9	1.4	1.1	0.9
ICE-S	2.9	1.5	1.0	0.8	0.7
ICE-W	5.1	3.0	2.1	1.8	1.5
LOB	5.1	2.9	2.1	1.6	1.4
SEU-S	3.2	1.8	1.3	1.0	0.9
SEU-W	5.3	3.3	2.5	2.2	2.0

Table 1.2: The OOV rates of various English corpora, spanning both written text (W) and transcribed speech (S), as a function of vocabulary size. The different-sized vocabularies are drawn from the British National Corpus. (From [Fang and Huckvale, 2000])

Named entities constituted 75% of OOV words and the rest were rare, compounded, or improvised words. It was also shown that OOV words caused 2 errors per occurrence.

In [Furui et al., 2005], an analysis of a large-scale continuous speech recognizer was reported on a corpus of spontaneous Japanese (CSJ) [Maekawa, 2003; Maekawa et al., 2004]. In the study, a regression model was proposed to model the recognition accuracy as a function of several parameters, one of them being the OOV rate. It was demonstrated that the recognition accuracy is highly correlated with the OOV rate, possibly due to the OOV ripple effect.

So far, we have motivated the need for robustly handling and learning OOV words. In the rest of this section, we present an overview of OOV modeling as well as sub-word modeling and L2S/S2L conversion.

1.1.2 Previous Work: OOV Modeling

The task of OOV modeling can be viewed as a two-stage process: (1) OOV word detection and (2) OOV word learning. In this section, we present an overview of the literature covering those two aspects of OOV modeling.

OOV Word Detection

Over the last two decades the interest in OOV detection has grown and resulted in a rich literature [Asadi et al., 1990; Asadi, 1991; Hayamizu et al., 1993; Suhm et al., 1993; Young, 1994a,b; Klakow et al., 1999; Bazzi and Glass, 2000a, 2001, 2002; Bazzi, 2002; Schaaf, 2001; Yazgan and Saraclar, 2004; Bisani and Ney, 2005; Thomae et al., 2005; Lin et al., 2007]. In this section, we present a sample of this work.

One of the earliest approaches to OOV detection was proposed in [Asadi et al., 1990; Asadi, 1991]. OOV modeling was achieved by introducing a generic OOV word to a word-based recognizer. The generic OOV word was modeled using hidden Markov models (HMM) [Rabiner and Juang, 1993], and different configurations were examined. This method is typically referred to as the *filler approach*, whereby the model absorbs the phonetic sequence corresponding to an OOV word. The HMMs were designed to generate any sequence of phones, but enforced a minimum number of 2 or 4 phones. In addition, both context-dependent and context-independent acoustic phone models were evaluated. The OOV models were tested on the DARPA 1000-word Resource Management database for continuous speech recognition [Price et al., 1988], and the Byblos BBN ASR system was used [Chow et al., 1987]. OOV words were artificially introduced by removing words from the lexicon. The OOV words were constrained to be in one of seven classes: ship name, ship name possessive, port name, water name, capability, land name and track name. The best performance was an OOV detection rate of 74% and a false alarm rate[4] of 3.4% for an HMM constrained to have at least 2 context-independent phonemes. These preliminary results demonstrated the challenges of robustly detecting OOV words even for small domain ASR and a highly constrained language model.

Multiple knowledge sources including acoustics, semantics, pragmatics, and discourse were combined to detect OOV words in limited-domain spontaneous dialogue systems in [Young, 1994a,b]. The evaluation was performed on subsets of ATIS [Price, 1990] using the SPHINX-I HMM-based ASR system [Lee et al., 1990]. Using a confidence measure derived from a normalized acoustic score, 53% of misrecognized words were accurately detected, and 6% of the correct words were mistakenly rejected. Next, a discourse model analysed the recognition output and its semantic parse, and was able to detect 88% of the contextually inconsistent errors. The combination of the acoustic confidence measure with the discourse model produced even better results where 73.1% of all misrecognized errors were correctly detected and 5% of the correct words were incorrectly rejected. The combined model detected 19% and 14% more errors than the acoustic and semantically-based systems alone respectively.

Extensive research on OOV modeling was presented in [Bazzi and Glass, 2000a, 2001; Hazen and Bazzi, 2001; Bazzi and Glass, 2002; Bazzi, 2002]. The research and the results summarised here were all reported on the JUPITER weather domain data [Zue et al., 2000] and were generated with the SUMMIT landmark-based ASR [Glass et al., 1996].

In [Bazzi and Glass, 2000a], a generic corpus-trained phone recognizer was implemented as an OOV model using the filler approach. Transition into the OOV model was controlled using a penalty term. Following the implementation of the phone-based OOV model, around 50% of OOV words were detected and very few in-vocabulary (IV) words were falsely classified as OOV words. Meanwhile, the WER of the ASR system suffered minor deterioration of 0.3%.

Three different configurations to the OOV model were investigated in [Bazzi and Glass, 2001]: (1) a phone-based oracle trained only on the OOV words in the test set, (2) a phone-based OOV model trained on the LDC Pronlex dictionary [Pronlex], and (3) an OOV model trained on variable-length units generated by combining phones using the mutual information (MI) criterion. The results showed that, when 6% of IV words were falsely rejected, the oracle OOV model had an OOV detection rate of around 87%, while the dictionary-based and MI OOV models had detection rates of

[4] False alarm rate refers to the frequency at which in-vocabulary words are classified as out-of-vocabulary.

around 74% and 80% respectively. The MI OOV model was able to outperform the initial results presented in [Bazzi and Glass, 2000a] and was the closest to the upper bound performance of the oracle model.

In [Hazen and Bazzi, 2001], the OOV detection model was followed by confidence scoring [Hazen et al., 2000a,b] on the hypothesized IV words. The combined approach reduced the false acceptance rate of misrecognized keywords by 25% while accepting 98% of correct keywords.

In [Bazzi and Glass, 2002], instead of implementing a single OOV model, several OOV models representing different classes of words were added. The classes were selected using two approaches: (1) part-of-speech tagging and (2) iterative and automatic clustering. In terms of results, there was no significant difference between the two approaches; however, there was gain in using more than one OOV model. For example, for an OOV detection rate of 70%, the false rejection of IV words was reduced from 5.3% for a single class to 2.9% for an eight-class model.

Further experiments were conducted on the JUPITER domain in [Bazzi, 2002], demonstrating that a dictionary-trained OOV model could detect 70% of the OOV words while falsely rejecting 5.3% of the IV words. At that operating point, the IV WER worsened by an absolute 0.3% while the overall WER improved by an absolute 0.7%. The OOV modeling experiments were also evaluated on the broadcast news domain, HUB4 [Graff and Liberman, 1997], which is less constrained and has a larger vocabulary than the JUPITER domain. The results had a similar trend to the ones obtained on JUPITER, but were worse overall. The overall WER on HUB4 improved from 24.9% without an OOV model to 23.5% with one.

A hybrid ASR system combining both words and sub-lexical units such as phones and syllables was proposed in [Yazgan and Saraclar, 2004]. Following recognition, the phone or syllable sequences in the output lattice were replaced with an OOV tag. OOV detection was then performed by modeling the OOV count in each utterance and comparing it to a threshold. On the other hand, the baseline system, which was a word-only ASR model, performed OOV detection using the utterance posterior probability. The assumption was that erroneous utterances would have low posterior probabilities. Experiments were conducted on the RT02 Switchboard Evaluation data [Garofolo et al., 2002] using the AT&T Switchboard Evaluation ASR system [Ljolje et al., 2002]. The results indicated that the hybrid model had a 10-15% improvement in OOV detection over the word-only model.

OOV Word Learning

In [Hetherington, 1995], it was proposed that learning a new word involves the update of three knowledge sources: the acoustic model, the lexicon, and the language model. In this section, we briefly present previous work related to each of these aspects.

Language model adaptation was discussed in [Jelinek et al., 1990], where new words were dynamically added to a statistical language model. The proposed approach avoided the need for a large amount of training data, by adding every newly encountered word to a *synonym* word class. A new and old word were synonymous if they had comparable word contexts. The resulting language model was evaluated by computing its perplexity[5] on several sets of text data. Perplexity results were also reported for new and old words separately. The results indicated that it was advantageous to use synonym word classes as opposed to a single generic new word class. Moreover, the synonym approach modeled new words well without compromising the performance of old words.

In [Asadi and Leung, 1993; Asadi, 1991], acoustic adaptation to new spoken words was examined. Adaptation was performed and evaluated on a limited data set consisting of the 25 most frequent cities in the metropolitan Boston area. The proposed approach required the orthographic spelling of new words, which was converted to a phonetic representation using letter-to-sound rules. The phonetic representations and the spoken instances of the new words were then used to adapt the acoustic model in a supervised fashion. The adapted acoustic model was evaluated on the city domain and results showed that, with around 30 tokens per new word, the word error rate was reduced from 34% to 8%. Two hundred tokens per new word reduced the error rate further to 4%. Although the results were significant, the research assumed the availability of the spelling of each new word as well as a reasonable amount of training data, which is not always the case.

Several approaches have been proposed to learn the pronunciation of a new word or its orthographic spelling or both [Asadi et al., 1991; Asadi, 1991; Suhm et al., 1993; Chung, 2000a,b, 2001; Galescu, 2003; Chung et al., 2004; Scharenborg and Seneff, 2005; Oger et al., 2008].

One of the earliest research efforts on automatic lexical learning was presented in [Asadi et al., 1991; Asadi, 1991]. Pronunciations were generated using a phonetic recognizer as well as DECtalk, a text-to-speech synthesizer [Hallahan, 1995]. The best results were obtained when both modules were combined. Phonetic transcriptions were initially generated by DECtalk, and were expanded into phonetic graphs using a phone confusion matrix. The phonetic graphs were then used to constrain the search space of a phonetic recognizer. The resulting pronunciations were comparable to manually transcribed ones and outperformed the pronunciations generated by a phonetic recognizer alone.

In [Scharenborg and Seneff, 2005], a 2-stage module was designed to handle OOV words in a continuous speech recognition task. OOV words were detected in the first stage using the filler approach described in [Bazzi and Glass, 2000a; Bazzi, 2002]. All words that were phonetically close to each OOV word were extracted from a fallback lexicon and added to the original lexicon.

A three-stage approach that can detect and learn OOV words was proposed in [Chung, 2000a,b, 2001]. The first stage, which encapsulated linguistic constraints [Seneff, 1996] and modeled both graphemic and pronunciation information, was used to generate the phonetic graphs. The second stage converted the phonetic graphs to word networks and identified possible OOV locations. In the third stage, the word networks were parsed using a natural language module [Seneff, 1992], and spellings

[5] Perplexity is a measure of how well a language model represents or models a text. It is a function of the entropy of the text. The lower the value of a language model perplexity, the better the language model.

were hypothesized for OOV words. The 3-stage system was evaluated on JUPITER [Zue et al., 2000], where the test utterances were chosen to have a single OOV word each. The OOV words were all city names. The results exhibited a 29% reduction in WER.

1.1.3 Previous Work: Sub-word Modeling and L2S/S2L Conversion

Sub-word modeling and L2S/S2L conversion go hand in hand, where L2S/S2L models are typically implemented at the sub-lexical level [Lucassen and Mercer, 1984; Alleva and Lee, 1989; Bahl et al., 1991; Meng et al., 1994a,b; Meng, 1995; Deligne et al., 1995; Fosler et al., 1996; Sloboda and Waibel, 1996; Westendorf and Jelitto, 1996; Deligne and Bimbot, 1997; Jiang et al., 1997; Black et al., 1998; Whittaker and Woodland, 2000; Kneissler and Klakow, 2001; Chung, 2001; Galescu and Allen, 2001; Bisani and Ney, 2002; Decadt et al., 2002; Galescu and Allen, 2002; Chen, 2003; Chung et al., 2004; Bisani and Ney, 2008]. Whereas sub-word modeling is concerned with modeling words using sub-lexical units, L2S/S2L modeling involves converting symbols from one domain to another (e.g. pronunciation to spelling). Moreover, both sub-word modeling and L2S/S2L conversion are critical building blocks in the process of learning the pronunciation and spelling representations of new lexical entries. Letter-to-sound models can typically be *inverted* to provide sound-to-letter (S2L) capabilities and vice versa. S2L transformations are useful to learn the graphemic representation of new words from phonetic transcriptions, whereas L2S models are commonly used for automatic lexicon learning and speech synthesis purposes.

[Bahl et al., 1991] was one of the first to model letter-to-sound and estimate phonetic baseforms from the model. Phonetic baseforms were learned using at least one spoken utterance of the word as well as automatically-derived L2S rules. The L2S rules were generated by aligning letter and phone strings based on context using decision trees. Context clustering was performed using decision trees, by posing binary questions about context, e.g. “Is the next letter a vowel?”. The phonetic representation was chosen to maximize the posterior probability of the pronunciation given the spoken utterance and the spelling. When faced with multiple utterances, the aforementioned approach was performed for each utterance and the generated pronunciations were concatenated. The generated phonetic baseforms were evaluated on an isolated word recognition task. The best results were obtained using four spoken utterances of each word and the corresponding spelling.

An unusual approach towards sound-to-letter conversion was proposed in [Alleva and Lee, 1989]: phonetic representation was completely bypassed, and HMMs were used to model letters of the alphabet instead of phones. To account for the highly context-dependent letters, each letter was modeled in the context of two left neighbors and one right letter. Silent letters, e.g. *g* and *h* in *night*, were handled by skipping entire HMMs. The sound-to-letter model was tested on 30 ship and place names and had a 39.3% letter error rate and a 21.1% word error rate.

The research in this thesis is influenced by the work presented in [Meng et al., 1994a,b; Meng, 1995] on bi-directional L2S/S2L modeling. The semi-automatic approach made use of a parser framework that modeled linguistic information in a hierarchical structure, which encoded morphology, stress patterns, syllabification, phonemics, and orthography. A set of hand-written rules defined the relations among the hierarchical layers. The parser was trained and tested on subsets of the Brown Corpus vocabulary [Kucera and Francis, 1967]. During L2S and S2L conversion, 6% and 5% of the input strings could not be parsed respectively. For the input that was parsed, the L2S model had a phonetic accuracy of 92.5%, and the S2L model had a letter accuracy of 89.4%.

In [Galescu and Allen, 2001, 2002], a bi-directional L2S model that incorporated grapheme-to-phoneme conversion units was designed using a joint *n*-gram model. Results were reported on the CMU dictionary [Weide, 1998] in terms of string accuracy - a spelling or phonetic transcription is accurate if it exactly matches the corresponding entry in the dictionary. The grapheme-to-phoneme accuracy was 71.5% and the phoneme-to-grapheme accuracy was 50%. The model was also tested on nouns only, and accuracies of 68% and 41% were reported for the grapheme-to-phoneme and phoneme-to-grapheme systems respectively.

In [Deligne et al., 1995], an unsupervised and statistical approach was devised to match multiple streams of symbols according to the maximum likelihood criterion. The resulting model, denoted as joint multi-gram, was trained on streams of phones and letters, where variable-length sequences of symbols from both streams were matched based on a maximum likelihood criterion. One of the outcomes of this research was a set of hybrid units that encoded both orthographic and pronunciation information. The joint multi-gram model proposed in [Deligne et al., 1995; Deligne and Bimbot, 1997] was repurposed for grapheme-to-phoneme conversion [Bisani and Ney, 2002, 2008]. In [Bisani and Ney, 2002], the joint multi-gram approach was used to generate hybrid units denoted graphones. A language model was then trained on the graphone alignments produced by the model. Experiments were performed on English and German phonetic transcription tasks. The phonetic error rates on the German task were lower than those on the English one. This is likely due to the simpler letter-to-sound rules in the German language. Different sized hybrid units were generated, and the best phonetic error rates, obtained with a maximum of two letters and two phones per unit, were 0.52% and 4.02% on the German and English lexicons respectively. In [Bisani and Ney, 2008], instead of separately implementing the joint multi-gram model followed by a language model as in [Bisani and Ney, 2002], the authors incorporated language modeling into the maximum likelihood training of the multi-gram model.

Other approaches towards letter-to-sound conversion and sub-word modeling include memory-based learning [Decadt et al., 2002], pronunciation by analogy [Marchand and Damper, 2000], and maximum entropy models estimated using decision trees [Chen, 2003].

1.2 Proposed Approach

In this research, we propose sub-word units as opposed to the conventionally used words as the basic lexical units in ASR. The sub-word units designed in this thesis encode only pronunciation information and can be considered agglomerations of one or more phonemes. Hybrid spellneme units that consist of combined graphemic and phonemic clusters are also designed in the process. A sub-word representation of the word `station` is:

```
station: st+ -ey+ shaxn
```

And the corresponding spellneme representation is:

```
station: st_st+ a_-ey+ tion_shaxn
```

The characteristic aspects of the sub-word and spellneme units are as follows:

- The sub-word units are designed using context-free rules that encode sub-syllabic linguistic knowledge such as positional and phonological and stress information.
- The spellnemes are automatically generated within a top-down parser framework, using the linguistically-motivated context-free rules.
- The spellnemes are derived using a technique that combines linguistic knowledge with statistical data driven methods, and as such, they differ from most of the previously proposed grapheme-to-phoneme and phoneme-to-grapheme units. However, this research is inspired by previous work [Meng, 1995; Chung, 2001].
- Given that the spellneme structure encodes both graphemic and phonemic information, it is straightforward to use it as a basic building block for designing bi-directional letter-to-sound models. This aspect is particularly useful for learning the spelling of a word given its phonetic representation, as well as automatically learning a lexical entry in a dictionary.

1.3 Contributions

The main contributions of this thesis are as follows:

The introduction of a novel L2S/S2L model: One of the major outcomes of the linguistically-motivated sub-word model are the spellnemes. The spellnemes are crucial for designing L2S and S2L models. In this thesis, we describe in detail the linguistically-motivated sub-word model, which was previously introduced in [Seneff, 1992, 2007]. We also describe the spellneme units, and we propose and implement a bi-directional L2S model using finite state transducers (FSTs), which map inputs to output strings through a parsimonious and efficient network representation.

In-depth investigation of automatic lexicon generation: We carefully assess the performance of the L2S and S2L models on the task of automatically learning lexical entries in a dictionary. In the first set of experiments, perfect knowledge of the spelling of new words is assumed. In the second set, a novel, unsupervised, and iterative approach is designed to learn both spelling and pronunciation of a new word from acoustics.

The evaluation of sub-word recognition and S2L on a lexical access task: We assess the ability of the S2L model to estimate the spelling of isolated words, and we evaluate its performance within a simple speech recognizer. The S2L model is plugged in as a back-end to a sub-word based back-off mechanism for a standard isolated word recognizer.

OOV word modeling and vocabulary compression for continuous ASR and information retrieval: The sub-syllabic sub-word units are evaluated in the context of OOV modeling in continuous ASR. The ASR system is embedded as a front-end for an information retrieval system that is accessed by spoken queries. OOV words are artificially introduced into the ASR system by reducing (*compressing*) its vocabulary size. A set of experiments tests: (1) how well the sub-word units can model new words; (2) how much the system vocabulary can be compressed without significant loss in recognition and retrieval performances.

1.4 Thesis Outline

The remainder of this thesis is organized into eight chapters. Figure 1-5 illustrates the thesis outline which is described below:

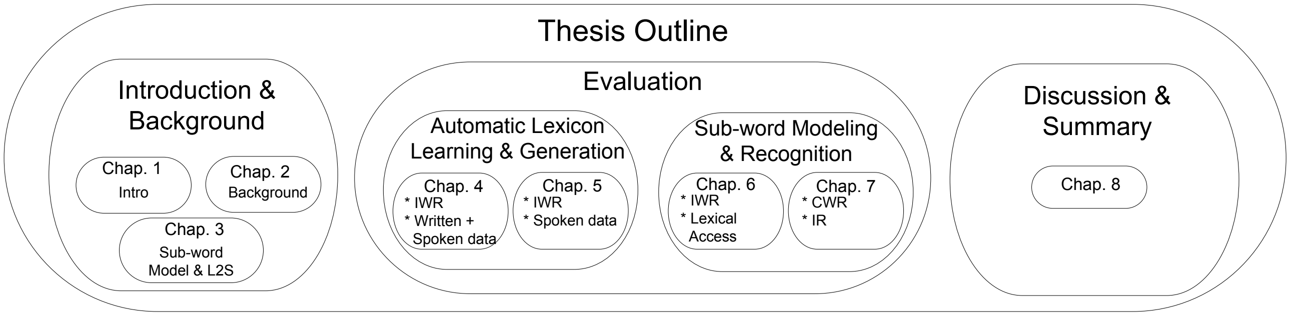


Figure 1-5: Thesis Outline (IWR = Isolated Word Recognition, CWR = Continuous Word Recognition).

- **Chapter 2: Background**
  We describe, SUMMIT, the landmark-based ASR system used in this research, as well as the types of OOV models typically implemented in the literature.

- **Chapter 3: The Linguistically-Motivated Sub-Word Model**
  We provide motivation and an overview of syllabic modeling in speech recognition. We describe the process of designing the sub-word and the spellneme units, using context-free rules within a parser framework supported by a probability model. We also present the implementation of the bi-directional L2S model using finite state transducers (FSTs).

- **Evaluation**

  - **Chapter 4: Automatic Lexical Pronunciation Generation and Update**
    We carefully assess the performance of the L2S model and the sub-word recognizer on the task of automatic lexical pronunciation generation. We propose two approaches for automatically generating lexical dictionaries:

    1. Using the **L2S model**, which maps letter sequences to phonetic pronunciation(s).
    2. Using the **sub-word recognizer**, whereby spoken instances of words are presented to the sub-word recognizer, generating sub-word sequences, which are then converted into phonetic representations.

    The research presented in this chapter assumes perfect knowledge of the spelling of the lexicon, which is inherently embedded in both approaches. The generated lexical dictionaries are evaluated in terms of Word Error Rate (WER) on an isolated word recognition task.

  - **Chapter 5: Turbo-Style Algorithm: An Unsupervised Approach Towards Lexical Dictionary Estimation**
    We pursue further the task of automatic lexical acquisition, and relax the assumption of perfect spelling knowledge. We propose an iterative and unsupervised algorithm, denoted Turbo-style, which presents spoken instances of both spellings and words to a letter and sub-word recognizer respectively, and fuses information from both systems to boost the overall lexical learning performance. The algorithm is evaluated in terms of spelling accuracy, letter error rate (LER), and phonetic error rate (PER) of the generated lexical entries. The automatically generated lexical dictionaries are also evaluated on an isolated word recognition task in terms of word error rate (WER).

  - **Chapter 6: A Hybrid Approach Towards Open-Ended Recognition Using Sub-Word Modeling**
    We embed the sub-word recognizer in an error recovery mechanism for an isolated word recognizer. The result is a parallel integration of word and sub-word recognizers, which is evaluated in a simple dialogue system. Users are prompted to speak a word and the word recognizer hypothesizes and displays the top candidate words. If the correct word is not in the returned list, the system backs off to the sub-word recognizer.

  - **Chapter 7: Recognition and Information Retrieval Experiments in the Lyrics Domain**
    We implement a song retrieval system, which is accessed via spoken lyrics. A flat hybrid ASR is designed as a front-end to the retrieval model by incorporating the sub-word units into the ASR lexicon and language model. The overall system is assessed in terms of recognition as well as information retrieval performance.

- **Chapter 8: Summary and Future Work**
  We conclude, summarise, and discuss the results and contributions of this thesis.

- **Appendix A: The Sub-Lexical Context-Free Grammar**

- **Appendix B: The Phonetic Alphabet**

- **Appendix C: Rhyme Splitting**

- **Appendix D: Sample Queries**

Chapter 2

Background

This chapter provides a description of the background relevant to this thesis. First we describe SUMMIT, the ASR system used in all the experimental setups in this research. This involves describing the acoustic, lexical, and language modeling as well as the finite-state transducer (FST) implementation of the speech recognizer. Next, we present the two types of OOV models commonly implemented in the literature.

2.1 The SUMMIT Speech Recognition System

The SUMMIT speech recognition system has been developed at the Spoken Language Systems Group at the MIT Computer Science and Artificial Intelligence Laboratory [Glass et al., 1996; Lee and Glass, 1998; Livescu and Glass, 2001; Glass, 2003]. Most current speech recognizers extract acoustic measurements over fixed-rate windows or frames [Bahl et al., 1983; Chow et al., 1987; Rabiner, 1989; Nguyen et al., 2005], and model the observation space using first-order hidden Markov models (HMM) [Rabiner, 1989; Rabiner and Juang, 1993]. SUMMIT, which is a landmark-based ASR system, processes these frames further to produce a network of variable-length segments, and models each segment with a fixed-size acoustic feature vector. In SUMMIT, the segments correspond to phones.

In general, the recognition problem can be mathematically modeled as follows. Given A, a set of acoustic observations corresponding to a speech waveform, the goal is to find the most likely sequence of words $W^* = w_1, w_2, \cdots w_N$ that satisfies the maximum a posteriori (MAP) criterion:

$$W^* = \underset{W}{\texttt{argmax}}\, P(W|A) \tag{2.1}$$

Where $W \in \mathcal{W}$, the set of all word strings, and $w_i \in \mathcal{V}$, a finite lexicon. Equation 2.1, can be rewritten as:

$$W^* = \underset{W}{\texttt{argmax}} \sum_{U,S} P(W, U, S|A) \tag{2.2}$$

Which now incorporates $S \in \mathcal{S}$, the set of all possible segmentations, and $U \in \mathcal{U}$, the set of all possible sub-lexical (phone) strings.

Similarly to other speech recognition systems, SUMMIT approximates the summation in Equation 2.2 by a maximization over segmentations and phone strings:

$$\begin{aligned} W^* &\approx \underset{S,U,W}{\texttt{argmax}}\, P(W,U,S|A) \qquad (2.3) \\ &= \underset{S,U,W}{\texttt{argmax}} \frac{P(A|S,U,W)P(S|U,W)P(U|W)P(W)}{P(A)} \\ &= \underset{S,U,W}{\texttt{argmax}} P(A|S,U,W)P(S|U,W)P(U|W)P(W) \end{aligned}$$

The right side of Equation 2.3 is a Bayes' rule expansion. $P(A)$ is typically ignored since it is not a function of S, U, or W. $P(A|S,U,W)$ corresponds to the acoustic model and $P(S|U,W)$ is a statistical model of the segmentation, more generally known as a duration model. In this thesis, $P(S|U,W)$ is kept constant. $P(U|W)$ is commonly known as the pronunciation model, and $P(W)$ is the language model.

In the rest of this section, we describe the various components of Equation 2.3. This includes the signal processing stage as well as the acoustic modeling process. We also provide a description of lexical and language modeling and the decoding process. Finally we describe the finite-state transducer (FST) implementation of SUMMIT.

2.1.1 Signal Processing

When a speech waveform is presented to SUMMIT, it is processed into a sequence of acoustic observation vectors. The most commonly used acoustic measurements are the Mel-frequency cepstral coefficients (MFCC) [Davis and Mermelstein, 1980]. MFCCs can be computed using a fast and efficient algorithm based on the fast Fourier transform computation. They also model well the non-linear frequency scale (Mel-scale) of the human auditory system. In SUMMIT, MFCCs are derived as follows:

1. For every 5ms, compute the short-time energy spectrum of an 8kHz speech waveform by calculating the magnitude squared of a 256-point discrete Fourier transform over frame intervals of width 25.6ms.

2. Multiply the energy spectrum by 40 triangular band-pass filters. The triangular filters are designed to incorporate a Mel-frequency warping with linear spacing below 1kHz and logarithmic spacing above that, and the frequency warping can be mathematically formulated as follows:

$$f' = 2595 \log_{10}(1 + \frac{f}{700})$$

   Compute the Mel-frequency spectral coefficients (MFSC), as the energy outputs of each filter.

3. Compute the log transform, $10 \log_{10}()$ of the 40 MFSCs

4. Take the discrete-cosine transform (DCT) of the logged MFSCs to whiten the MFSC space and project it onto a 14-dimensional space.

$$MFCC[i] = \sum_{k=0}^{K-1} \cos \frac{\pi i(k-1/2)}{K} MFSC_{log}[k] \quad m = 0 \ldots M, K = 40$$

By applying the DCT transform, the MFSC cofficients are decorrelated and the MFCCs can be modeled efficiently with diagonal Gaussian mixture models instead of full covariance ones.

2.1.2 Segmentation

Once each speech frame is converted into a 14-dimensional MFCC vector, a segment network is produced by hypothesizing acoustic landmarks or boundary locations. In SUMMIT, the network is a graph of phonetic labels and their associated scores. The phonetic scores represent the confidence of the network in the segmentation as well as in the phonetic accuracy.

Major landmarks are hypothesized at locations where the spectral change exceeds a global threshold. Minor landmarks are also detected between major landmarks at locations where the spectral change exceeds a local threshold. The minor landmarks are fully interconnected within but not across major landmarks. On the other hand, each major landmark is connected to its two right adjacent major landmarks. The reader is referred to the following sources for further detail on the SUMMIT segmentation process [Glass and Zue, 1988; Glass, 1988; Zue et al., 1989b].

Figure 2-1 illustrates a segment network corresponding to the utterance *computers that talk*, and focuses on two (shaded in gray) possible segmentations through the graph.

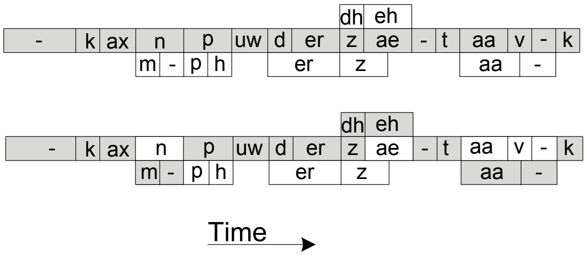


Figure 2-1: An illustration of a portion of the segment network for the utterance *computers that talk*. The figure depicts two possible segmentations, shaded in gray. The white segments correspond to units with no phonetic representation either because they were too short or too long. (From [Glass, 2003])

In the next section we discuss the challenges of acoustically modeling the network observation space compared to the conventionally used frame-based recognition models.

2.1.3 Graph-based Observations

Frame-based ASR typically computes temporal sequences of spectral observations at a fixed rate. The observation space, A in Equation 2.1, is the whole observation sequence, and the likelihood term $P(A|S,U)$ can be directly compared over competing words. On the other hand, in SUMMIT, the observation space, A, consists of all the observation vectors in a segment network. So if we consider the set X of observations corresponding to the shaded segments of a particular segmentation in Figure 2-1, then the term $P(X|S,U)$ is not comparable across different segmentations. In fact, there is a need to consider all the observations in the network. One approach to dealing with this problem is to introduce the set Y of all segments corresponding to the non-lexical units (the white segments in Figure 2-1), such that $X \cap Y = \emptyset$ and $X \cup Y = A$. The set Y can be modeled with the anti-phone unit, $\overline{\alpha}$ as proposed in [Glass et al., 1996; Chang and Glass, 1997; Chang, 1998] and illustrated in Figure 2-2.

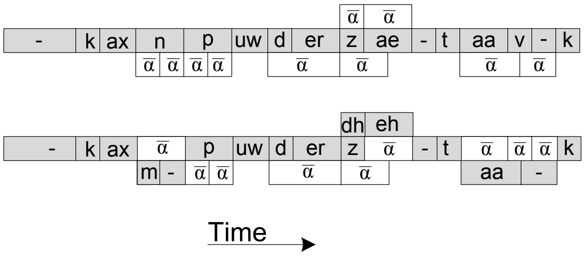

$\overline{\alpha}$ $\overline{\alpha}$
- k ax n p uw d er z ae - t aa v - k
$\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$
dh eh
- k ax $\overline{\alpha}$ p uw d er z $\overline{\alpha}$ - t $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ k
m - $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ $\overline{\alpha}$ aa -
Time

Figure 2-2: A replicate of Figure 2-1 with every white segment corresponding to a non-lexical unit in a particular segmentation replaced by the anti-phone unit $\overline{\alpha}$. (From [Glass, 2003])

In this case, the term $P(A|S,U)$ can be modeled as follows:

$$\begin{aligned} P(A|S,U) &= P(X,Y|S,U) \\ &= P(X|U)P(Y|U) \\ &= P(X|U)P(Y|\overline{\alpha})\frac{P(X|\overline{\alpha})}{P(X|\overline{\alpha})} \\ &\propto \frac{P(X|U)}{P(X|\overline{\alpha})} \end{aligned} \tag{2.4}$$

In the process of deriving Equation 2.4, the following assumptions are made: (1) X and Y are conditionally independent given U in the second line, and (2) $P(Y|U)$ depends only on $\overline{\alpha}$ in the third line and can, hence, be rewritten as $P(Y|\overline{\alpha})$. In the fourth line of Equation 2.4, the term $P(Y|\overline{\alpha})P(X|\overline{\alpha}) = P(X,Y|\overline{\alpha})$ is ignored since it is constant for any network.

2.1.4 Acoustic Modeling

In this thesis, instead of deriving measurements over each segment, we compute landmark or boundary observations at every hypothesized acoustic landmark. In this particular case, the landmark observations account for all the acoustic space, A, and there is no need for normalization as was the case for segmental observations in the previous section.

As illustrated in Figure 2-3, a telescopic MFCC average is extracted over 8 regions surrounding the boundary to create a 112-dimensional feature vector. Principal component analysis is used to reduce the correlation as well as the dimensionality (from 112 to 50) of the feature vector [Duda et al., 2000]. The 50-dimensional vectors are then used to train context-dependent diphone models which are modeled using diagonal Gaussian mixtures with a maximum of 75 mixtures per model.

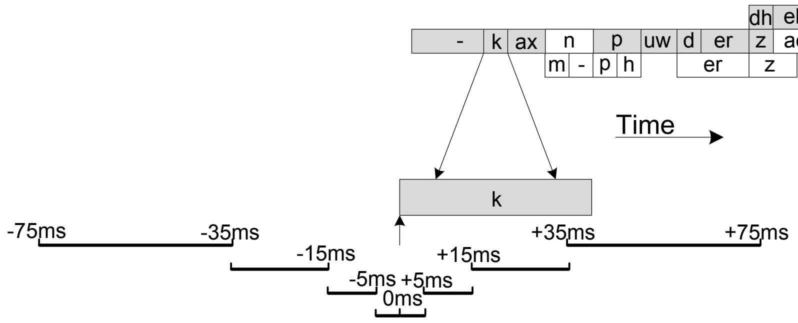


Figure 2-3: The boundary measurement extracted at the landmark indicated by the arrow. The measurement is a telescopic MFCC average derived over 8 regions surrounding the boundary, and is computed at every hypothesized landmark.

2.1.5 Lexical Modeling

A lexical model or a lexicon maps a set of words to their pronunciations. In SUMMIT a pronunciation is represented in terms of a string of phonemes. In addition to modeling one or more pronunciations for each word in the vocabulary, SUMMIT incorporates phonological rules that model phonetic variations of phonemes such as assimilation, deletion, and insertion [Zue et al., 1990; Hetherington, 2001]. Phonological rules are designed by lexical experts and applied automatically to the lexicon in order to generate alternative pronunciations.

2.1.6 Language Modeling

In this research, the language model (LM) is implemented as an n-gram [Manning and Schutze, 1999], which captures the statistical properties of sequences of n words. An n-gram makes the assumption that a word w_i is only dependent on the previous $n-1$ words, w_{i-1}, w_{i-2}, $\cdots$, w_{i-n}. Taking this assumption into consideration, the probability of a sequence of M words, W, is formulated as follows using the chain rule:

$$P(W) = \prod_{i=1}^{M} P(w_i|w_{i-n+1}, \cdots, w_{i-1}) \tag{2.5}$$

To avoid assigning zero probability to unseen n-grams, smoothing or discounting techniques are typically employed to redistribute probabilities from seen to unseen n-grams [Manning and Schutze, 1999; Chen and Goodman, 1996]. In this thesis, we build word, sub-word, and phoneme LMs of orders n = 2, 3, ... ,7. We use expectation maximization smoothing to estimate the probabilities of unseen n-grams [Baum, 1972].

2.1.7 Decoding

During decoding the recognition space is searched for the word sequence with the best score. The search space is created by scoring the segment network with the acoustic model and combining the result with the lexical constraints and the LM scores. SUMMIT performs efficient decoding using a two-pass approach. In the forward pass, the search space is pruned using a beam search, and Viterbi search is used to compute the best score [Soong and Huang, 1991]. The backward search is implemented using A^*, a best-first search that uses a distance-plus-cost heuristic function [Nilsson, 1980]. The distance-plus-cost function is the sum of two scores: (1) the actual lowest score from the source to the current node and (2) a heuristic estimate of the score from the current to the goal node. The heuristic estimate is obtained from the Viterbi intermediate scores derived in the forward path. During the Viterbi forward search a low-order n-gram, typically bigram, is applied. During the backward A^* search, scores from a higher-order n-gram can be incorporated.

2.1.8 Finite-State Transducer Implementation

The SUMMIT search space is implemented as a weighted finite-state transducer (FST) [Mohri, 1997; Hetherington, 2004]. FSTs have the ability to model transformations from one domain to another, e.g. words to phonemes as illustrated in Figure 2-4, as well as incorporate statistical knowledge in the form of weights.

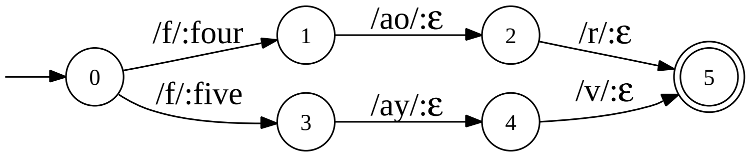


Figure 2-4: An FST that maps an input phonemic alphabet, I = {/f/, /ɔ/, /ɑʸ/, /v/ to an output word-based alphabet, O = {four, five}. ε denotes the null symbol, indicating, in this case, that no output symbol is emitted. This example is an FST implementation of a lexicon containing only the words *four* and *five.*

The SUMMIT search space is represented as a weighted FST, R, which is the composition of four FSTs:

$$R = C \, o \, P \, o \, L \, o \, G \tag{2.6}$$

Where C denotes the mapping from context-dependent model labels to context-independent phone labels, P the phonological rules that map phone labels to phoneme sequences, L the mapping from phonemes to words, and G the grammar or LM. A search through R produces a 1-best output, an N-best list, or a graph. Algorithms for FST optimization and efficient search have made FSTs an attractive framework for speech recognition [Mohri, 1997].

2.2 Out-of-Vocabulary Models

Out-of-vocabulary (OOV) modeling typically involves some form of sub-lexical representation of new words. The difference between various types of OOV models often lies in how this sub-lexical knowledge is integrated into an ASR. In this section, we discuss two OOV models commonly implemented in the literature.

2.2.1 The Hierarchical Filler OOV Model

In hierarchical filler OOV models, the ASR lexicon is augmented with one or more OOV tags. Typically, a single OOV symbol is used to represent all new words [Asadi et al., 1990; Bazzi and Glass, 2000a; Scharenborg and Seneff, 2005], but researchers have investigated multiple OOV classes that model different types of words, such as nouns, verbs, and adverbs [Bazzi and Glass, 2002]. Each OOV symbol is modeled with a sub-lexical network of phones, or syllables, etc. The underlying network can be viewed as a sub-lexical recognizer that can hypothesize any possible string of sub-word units. The filler model is denoted as hierarchical because the sub-lexical network is embedded in a large-scale ASR which is guided by a word-based LM, and the network is triggered only when the OOV symbol is hypothesized. This concept is illustrated in Figure 2-5, where the ASR system, guided by an m-gram, hypothesizes an OOV word with probability $P(OOV|word_{i-1}, word_{i-2}, \ldots, word_{i-m+1})$. When the OOV model is triggered, a sequence of sub-word units is generated guided by a sub-word n-gram, $P(sub\text{-}word_i|sub\text{-}word_{i-1}, sub\text{-}word_{i-2}, \ldots, sub\text{-}word_{i-n+1})$. A filler model can be utilized simply for OOV detection, or the sub-word representation generated by the network can be further processed to learn the pronunciation and spelling of a new word. In the process of designing a filler OOV model, it is necessary to tune the penalties for transitioning into and out of the OOV model. The penalty parameters affect whether the OOV symbol is being over- or under-generated, and whether the OOV model is absorbing an adequate number of sub-lexical units. The reader is referred to the following literature for further information on the filler OOV model [Bazzi and Glass, 2000a, 2001, 2002; Bazzi, 2002; Asadi et al., 1990; Asadi, 1991].

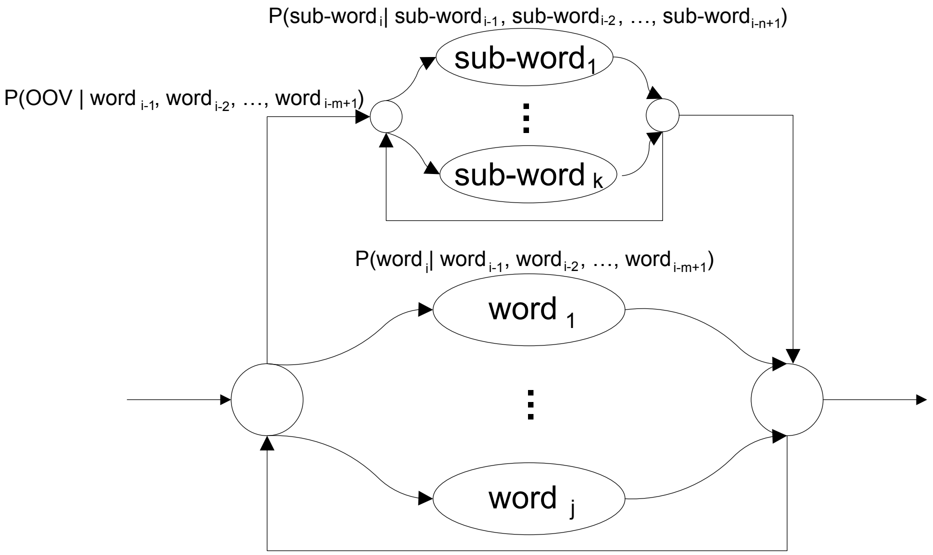

P(sub-word$_i$| sub-word$_{i-1}$, sub-word$_{i-2}$, ..., sub-word$_{i-n+1}$)
P(OOV | word $_{i-1}$, word$_{i-2}$, ..., word$_{i-m+1}$)
sub-word$_1$
sub-word $_k$
P(word $_i$| word$_{i-1}$, word$_{i-2}$, ..., word$_{i-m+1}$)
word $_1$
word $_j$

Figure 2-5: A filler OOV model embedded in a word-based ASR. During decoding, the ASR system hypothesizes an OOV word with probability $P(OOV|word_{i-1}, word_{i-2}, \ldots, word_{i-m+1})$ (guided by an m-gram). Upon entering the OOV model, a sequence of sub-word units is generated guided by a sub-lexical language model, which is implemented as an n-gram, $P(sub\text{-}word_i|sub\text{-}word_{i-1}, sub\text{-}word_{i-2}, \ldots, sub\text{-}word_{i-n+1})$.

2.2.2 The Flat Hybrid OOV Model

To implement a flat hybrid OOV model, the ASR lexicon is augmented with the sub-lexical units, and the designated OOV words in the LM are replaced with their sub-lexical representation. The result is a hybrid ASR system capable of hypothesizing sequences of both words and sub-word units. A hybrid OOV model is illustrated in Figure 2-6, where the ASR system hypothesizes either a word or a sub-word unit guided by a hybrid language model. The language model is implemented as an n-gram, $P(c_i|c_{i-1}, c_{i-2}, \ldots, c_{i-n+1})$, where c_i can be a word or a sub-word. Whereas the filler model integrates two separate - word and sub-word - recognizers, the flat hybrid model combines the word and sub-word units into a single recognition space. The model is denoted flat since it is capable of predicting and modeling OOV words simultaneously guided by a hybrid LM which contains both words and sub-word units. The accuracy of the flat model in detecting and modeling OOV words is correlated with the LM hybrid training data and the associated frequency of OOV words: the fewer the OOV words in the LM training data, the less likely it is that the hybrid ASR will generate sub-word sequences. Similarly to the filler model, a flat hybrid OOV model can both detect an OOV word as well as model its spelling and pronunciation. Previous work in the literature, which have successfully implemented flat hybrid OOV models are [Galescu, 2003; Yazgan and Saraclar, 2004; Bisani and Ney, 2005].

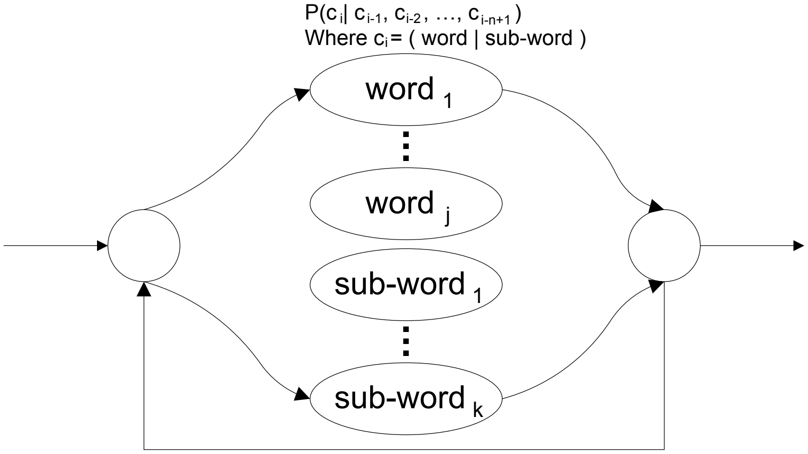

P(c_i| c_i-1, c_i-2, ..., c_i-n+1)
Where c_i = (word | sub-word)
word 1
word j
sub-word 1
sub-word k

Figure 2-6: A flat hybrid OOV model with a lexicon consisting of both words and sub-word units. During decoding, the ASR system hypothesizes either a word or a sub-word unit guided by a hybrid language model. The language model is implemented as an n-gram, $P(c_i|c_{i-1}, c_{i-2}, \ldots, c_{i-n+1})$, where c_i can be a word or a sub-word.

Chapter 3

The Linguistically-Motivated Sub-Word Model

In this chapter, we propose a framework for modeling sub-lexical knowledge using syllabic and sub-syllabic units. The proposed units are designed primarily using a context-free grammar which encapsulates phonotactic contraints and predominant stress patterns in the English language. In the first section, we motivate our use of syllable-inspired units by discussing the syllable as a structural unit for phonological representation. We present some background relating to the syllable theory, and we cover a brief overview of the role of syllables in speech recognition. Next, we describe in detail the model used to design the linguistically-motivated sub-word units. The linguistic model and the engineering framework, which were introduced in [Seneff, 1992; Seneff et al., 1992; Seneff, 2007] make use of context-free rules within a parser framework. Finally we present our bi-directional letter-to-sound model which is designed using hybrid units denoted as spellnemes. The spellnemes encode spelling and pronunciation knowledge, and are generated by the sub-word model presented in this chapter.

3.1 The Syllable

3.1.1 Background

It turns out the answer to the question: “What is a syllable?” is not as straightforward as one might think. In 1975, Ladefoged summarized the complexity of this question by stating [Ladefoged, 1975]:

> Although nearly everybody can identify syllables, almost nobody can define them. If I ask you how many syllables there are in “minimization” or “suprasegmental” you can easily count them and tell me. In each of these words there are five syllables. Nevertheless, it is curiously difficult to state an objective measure for locating the number of syllables in a word or a phrase (p.218).

Not only is the syllable hard to define formally, but its role in phonological theory was debatable. For example, generative phonology, which was introduced by Chomsky and Halle [Chomsky and Halle, 1968], models phonological representations as linear sequences of feature bundles denoted as segments. Features are associated with speech articulators and perception. The generative phonology framework proposed the segment as a structural unit and totally dismissed the syllable. However, several phonologists have identified the syllable as a critical linguistic unit, and argued that only by reference to the syllable structure can phonological aspects such as phonotactic constraints, stress, and tone be explained [Kahn, 1976; Hulst and Smith, 1082, 1982; Clements and Keyser, 1983]. In [Hulst and Smith, 1982], the syllable is described as a hierachical structural unit of the form shown in Figure 3-1[1]. The first division splits the syllable into an onset (the initial consonant cluster) and a rhyme (the rest). The rhyme can be further split into a nucleus and a coda (the final consonant cluster). A syllable is deemed well-formed if it satisfies the *Sonority Sequencing Principle*, which states that, within a syllable, there exists a sonority peak that is preceded and/or followed by segments with decreasing sonority value. A sonorant is a sound produced with the vocal tract excitation at the glottis and little constriction in the vocal tract. A sonorant scale, such as the one proposed in Table 3.1, ranks sounds according to their sonority based on how voiced they are and the level of constriction in the vocal tract. As indicated in Table 3.1, vowels are the most sonorant sounds, and stops are the least. The segment with the highest sonority scale is assigned to the nucleus of a syllable.

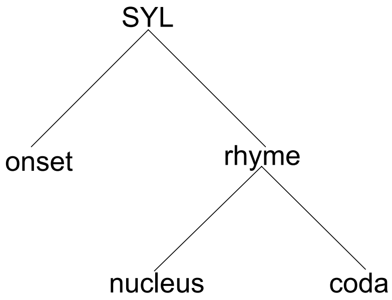


Figure 3-1: A tree representation of the hierarchical structure of the syllable. A syllable is typically decomposed into an onset and a rhyme. A rhyme can be further split into a nucleus and a coda.

Using the knowledge that each syllable should contain a sonority peak, syllabification can be performed through a set of principles such as the *Maximum Onset Principle* and the *Re-syllabification Principle.* The Maximum Onset Principle states that in a syllable, the onset should include as many consonants as allowed by the language structure (e.g. `arcane` ⟶ `ar cane` as opposed to `arc ane`). The Re-syllabification Principle states that a consonant cluster should be reassigned to the

[1] In [Hulst and Smith, 1982], the term *nucleus* is replaced by *peak.*

Sounds	Sonority Scale	Examples
low vowels	10	/ɑ/, /ɔ/
mid vowels	9	/ə/, /o/
high vowels	8	/i/, /u/
flaps	7	/ɾ/
laterals	6	/l/
nasals	5	/m/, /n/, /ŋ/
voiced fricatives	4	/v/, /ð/, /z/
voiceless fricatives	3	/f/, /θ/, /s/
voiced stops	2	/b/, /d/, /g/
voiceless stops	1	/p/, /t/, /k/

Table 3.1: A proposed sonority scale used to rank phonological segments. The sonority ranking allows the definition of well-formed syllables. Vowels have the highest sonority rank while stops have the lowest. (From [Randolph, 1989])

rhyme of the preceding syllable if that syllable is stressed (e.g. `actor` ⟶ `act or` as opposed to `ac tor`).

In addition to the emergence of various phonological theories that promoted the syllable as an essential linguistic unit, several influential doctoral theses examined the role of the syllable in phonological representations [Kahn, 1976; Church, 1983; Randolph, 1989]. In [Kahn, 1976], the shortcomings of generative phonology [Chomsky and Halle, 1968] were addressed. In particular, the author argued that by using syllabic analysis certain phonological processes, such as /t/-flapping and /r/-insertion and deletion, can be accounted for. In [Church, 1983], a bottom-up hierarchical approach was used to parse a phone sequence into a string of syllables using context-free phrase-structure rules. The framework proposed in [Church, 1983] was the first to use context-free grammars to model sub-lexical knowledge and inspired the research presented both in [Chung, 2001] as well as in this thesis. In [Randolph, 1989], an extensive empirical study was presented on the role of the syllable in modeling allophones[2] of English stop consonants. Regression trees were successfully used to predict the allophonic realisation of a stop consonant from several contextual factors, including the location of the stop in the syllable.

3.1.2 The Syllable in Speech Recognition

The general trend in speech recognition has been to decode a speech utterance into a string of phonemes, which is then mapped to a sequence of words [Chow et al., 1987; Zue et al., 1989b; Glass, 2003; Lamere et al., 2003; Prasad et al., 2005; Nguyen et al., 2005]. However, over the last three decades, some researchers have moved away from phonemes towards syllables, which are larger linguistically-motivated structural

[2] Allophones are different acoustic realizations associated with the same phoneme. For example the /t/ in *top* is aspirated, in *stop* is unaspirated, and in *butter* is flapped. All three sounds are allophones of /t/.

units capable of capturing phonotactic contraints and higher-level prosodic knowledge. Most of the previous work focused on incorporating syllables into a speech recognizer from an acoustic modeling perspective. In this thesis, the designed syllabic and sub-syllabic units are incorporated into the *lexical model* and/or the *language model* of an ASR rather than the acoustic model. The units are phonemic in the sense that they can be viewed as clusters of phonemes. However, the units are independent of the underlying acoustic representation adopted by the speech recognizer.

One of the earliest proponents of the syllable as a basic unit for speech recognition is [Fujimura, 1975]. The author argued that the phoneme exhibits too many irregularities in its phonetic realizations and a recognition unit at least the size of a syllable is required to capture acoustic variations such as those introduced by co-articulation[3]. This argument was further explored and supported by [Greenberg and Kingsbury, 1997; Hausenstein, 1997; Wu et al., 1998a; Ostendorf, 1999]. In [Greenberg and Kingsbury, 1997], the authors suggested that syllables are the basic sound units of speech both at the acoustic and the lexical level. Moreover, they proposed a new spectral representation denoted as *The Modulation Spectrogram*, which highlights syllabic nuclei and which is more robust to noise than the more traditional narrow-band spectrogram. In [Wu et al., 1998a], the use of syllable-like units for speech recognition was further motivated through the concept of *echoic memory* - the brief mental echo that continues to sound after an auditory stimulus. It was argued that, since the perceptual buffer associated with human echoic memory can store around 250 ms of sound, and since 80% of syllables have a duration of 250 ms or less, then the syllable is the largest unit of sound which can be stored in the echoic memory. This observation further motivated syllables as the obvious units for speech segmentation and recognition.

Syllables and syllable-like units have been sucessfully implemented as recognition units in [Schukat-Talamazzini et al., 1992; Hu et al., 1996; Hausenstein, 1997; Jones et al., 1997; Pfau et al., 1997; Bazzi and Glass, 2000b; Chung, 2000a,b, 2001; Ganapathiraju et al., 2001; Zhang, 2005; Han et al., 2006].

In [Schukat-Talamazzini et al., 1992], syllable-like units denoted as context-freezing units were shown to perform comparably to context-dependent phones. In [Hu et al., 1996], speech was segmented into syllable-like units by combining phoneme sequences for which the boundary is difficult to detect. The results were better than those obtained for a phoneme-based segmentation. Syllable-based HMM models were examined in [Jones et al., 1997], and compared to a phoneme-based recognizer that used monophone acoustic models and a bigram language model. The authors reported significant recognition improvement using the syllables, though they acknowledged the unfairness of comparing syllables to monophones instead of triphones. The performance of phones and syllables in a two-stage recognizer was evaluated and compared to a single-stage word recognizer in [Bazzi and Glass, 2000b]. The two-stage process consisted of a sub-lexical recognizer, in this case a phone or syllable recognizer, followed by a mapping from sub-lexical units to words. The results showed that the

[3] Co-articulation refers to the overlapping motion of articulators (e.g. lips, tongue), which are associated with adjacent articulations, and it is a common phenomenon in spontaneous speech.

syllable-based system outperformed the phone-based one even when high-order phone n-grams were used. However, both phone and syllable-based recognizers were unable to outperform the word-based recognizer.

The research presented in this thesis is inspired by the work presented in [Seneff, 1996; Chung, 2001], where a hierachical sub-lexical model is designed bottom-up using a context-free grammar. However, this research is based on a top-down parser that encodes pronunciation in pre-terminal units, and encodes all the spelling variants in the terminals. The use of a top-down parser to model the sub-syllabic structure of words is motivated by a much simpler notation scheme which ties directly to a phoneme notation typically used in phoneme-based speech recognizers. Another benefit of this parser is the ability to leverage from its tools which allow the conversion of bi-directional letter-to-sound models into finite state transducers (FST) that can be easily integrated within an FST-based recognizer [Glass, 2003].

Recognizers that combine syllable and phone-based knowledge can potentially yield better performance than systems incorporating only one of these knowledge sources, as demonstrated in [Wu et al., 1998a,b; Sethy et al., 2003]. In [Wu et al., 1998b], the syllable and phone-based systems were combined at the frame, syllable, and utterance levels. The context window over which acoustic measurements were extracted was increased from 105ms for phones to 185ms for syllables. The acoustic measurement was based on the modulation spectrogram proposed in [Greenberg and Kingsbury, 1997]. All three types of system integrations exhibited a superior performance over the phone baseline for both clean and reverberant speech. A mixed syllabic-phonetic system was proposed in [Sethy et al., 2003], where entries in the lexicon were modeled in terms of hybrid syllable and phoneme sequences. The hybrid system was evaluated on heavily accented and spontaneous speech and shown to outperform a contemporary state-of-the-art phone-based recognizer.

More recently, an ASR system was augmented with knowledge of the syllable nucleus position and count in [Bartels and Bilmes, 2008]. In the oracle experiments, the syllable nucleus count was determined by counting the number of vowel sounds in each word. Next, the syllable location and count were estimated within a Dynamic Bayesian Network framework. The results for the oracle system indicated that there is benefit in modeling the location of the syllable nucleus. However, further research needs to be done on reliable estimation of the syllable nucleus location before significant improvement can be observed.

3.2 The Linguistic Model

3.2.1 The Model Structure

This thesis uses the linguistic model introduced in used [Seneff, 2007], which is based on the English syllable structure. Since the whole syllable is deemed too large to generalize to unseen data, the syllable is primarily decomposed into an onset and rhyme as previously illustrated in Figure 3-1. The onset and rhyme are associated with sub-word units which encode pronunciation, and which could be used as pronunciation units in a lexical dictionary. A separate sub-word lexicon file defines the phonemic representation of the sub-word units.

The linguistic model is specified via a context-free grammar (CFG), which defines sub-syllabic structure, and which is designed through an iterative process. First, a small *seed* phonemic lexicon is converted into a sub-word representation resulting in an initial set of labeled training data. The labeled data are parsed with the grammar constrained by a filter that enforces the sub-word sequence provided for each word. Any parse failures are attributed to either missing or inaccurate grammar rules or to sub-word baseforms errors in the training data. Guided by the parse failures, manual edits are introduced into the grammar rules or the sub-word baseforms. This process is iterated until no parse failures are recorded.

The grammar is supported by a probability model, which is automatically trained on a set of parsed training data. The probability model is specified to capture the statistics of a node in the parse tree conditioned on its parent and its left sibling, and hence, encodes spacio-temporal context. Once the probability model is sufficiently trained and has built up considerable knowledge of the syllable structure, it can guide the grammar in parsing new words. This whole process can be employed to incrementally parse a large lexicon starting from a small set of labeled data as illustrated in Figure 3-2. By parsing the large lexicon, an alignment is automatically generated between the letters and the sub-word units corresponding to every lexical entry.

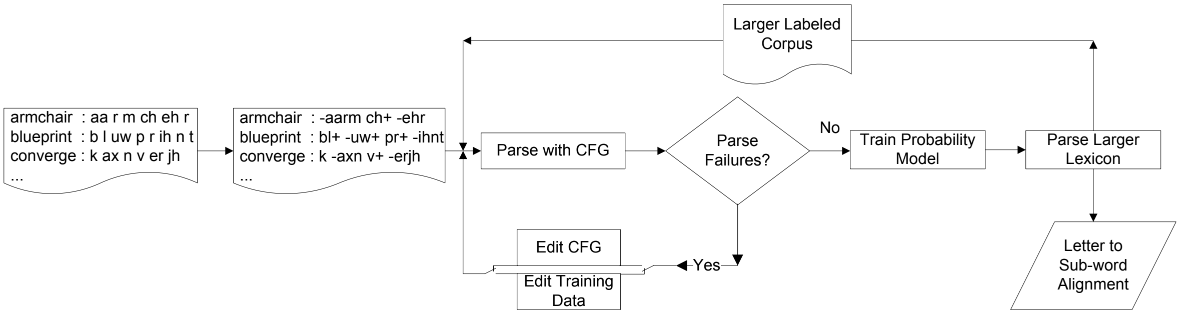

Larger Labeled Corpus
armchair : aa r m ch eh r
blueprint : b l uw p r ih n t
converge : k ax n v er jh
...
armchair : -aarm ch+ -ehr
blueprint : bl+ -uw+ pr+ -ihnt
converge : k -axn v+ -erjh
...
Parse with CFG
Parse Failures?
No
Train Probability Model
Parse Larger Lexicon
Edit CFG
Edit Training Data
Yes
Letter to Sub-word Alignment

Figure 3-2: The boot-strapping approach that is adopted to design the context-free rules and train the probability model. Starting with a small seed sub-word baseforms file, the labeled data are incrementally built while fine-tuning the grammar and the probability model. The final outcome of this iterative procedure is an alignment between letters and sub-word units for every entry in the lexicon. This alignment is used to train a statistical letter-to-sound/sound-to-letter model.

The ultimate goal of the linguistic model proposed in this research is, in fact, to obtain a high-quality alignment between graphemic and sound units, which would result in hybrid units that encode spelling and sound knowledge. These hybrid units, which are denoted as spellnemes, are used to design a letter-to-sound/sound-to-letter model. The alignment between graphemic and sound units is guided by the linguistically-motivated context-free grammar (CFG) and the parser framework. The grammar describing the sub-syllable structure and the underlying parser framework have been previously introduced in [Seneff, 1992; Seneff et al., 1992; Seneff, 2007]. The grammar design is covered in more detail in Section 3.2.2, where we also present two significant outcomes of the design: (1) the sub-syllabic sub-word units and (2) the spellnemes, which are the hybrid units encoding graphemic and phonemic information. In Section 3.2.3, we describe the probability model superimposed on the context-free grammar, and in Section 3.2.4, we briefly present the underlying parser framework.

3.2.2 Previous Work: The Grammar

The context-free grammar (CFG) used in this research has been designed to encode positional and phonological constraints in the English sub-syllabic structure [Seneff, 2007]. The decision to represent sub-syllabic as opposed to whole-syllabic structure is motivated by the hypothesis that the former would generalize better to unseen data. In [Fujimura, 1975], it was recommended that syllables be classified in terms of classes of features such as the nucleus, and that stressed and unstressed syllables be distinguished. These recommendations are incorporated in our sub-word design, as will be shown in this section.

Figure 3-3 illustrates the parse tree obtained for the phrase *copyright infringements* with the designed grammar. The root of the parse tree is `WRD`, and the hierarchical structure below the root consists of three layers. The second layer describes the sub-syllabic structure, primarily consisting of the `onset` and `rhyme`, as illustrated in Figure 3-1. The grammar makes use of sonority rules within a syllable combined with the Maximum Onset Principle described in Section 3.1.1 to make informed decisions about syllable boundary locations. Apart from `onset` and `rhyme`, several linguistically-motivated categories are introduced in the CFG to account for exceptions and special cases in the English language. For example, **pre** models certain unstressed prefixes as illustrated in Figure 3-3(b) for the word *infringements*. **ambi** which denotes *ambisyllabic*, is introduced for a subset of intersyllabic consonants to allow ambiguity in the syllable assignment. The `ambi` structure models the flapped-/t/ in Figure 3-4 for the word *attic*. The **affix** category models mostly coronal consonants which violate sonority rules in the coda as discussed earlier in Section 3.1.1. For example, according to Table 3.1, $SonorityScale(/\mathtt{t}/) < SonorityScale(/\mathtt{s}/)$. Hence, the /s/ in *infringements* in Figure 3-3(b) violates the sonority rule in the coda and is assigned to the structure `affix`. **usyl**, which stands for *unstressed syllable*, denotes a set of combined onsets and rhymes that form frequently occurring unstressed syllables such as `maxnt` in Figure 3-3(b). Finally, the first stressed `onset` and `rhyme` are distinguished from the rest of the categories and are represented by the suffix 1.

We illustrate below some sample rules from the second layer of the CFG, which define the sub-syllabic structure of English words:

```
WRD  ⟶ onset1 rhyme1 [usyl] rhyme ( usyl affix )
WRD  ⟶ onset1 pre rhyme1
WRD  ⟶ [pre] [onset1] rhyme1 usyl [ambi] rhyme [affix]
WRD  ⟶ [onset1] rhyme1 usyl [affix] onset [usyl] rhyme [rhyme]
WRD  ⟶ [onset1] rhyme1 ( ambi onset ) usyl [affix] rhyme
        ( ambi onset ) rhyme
```

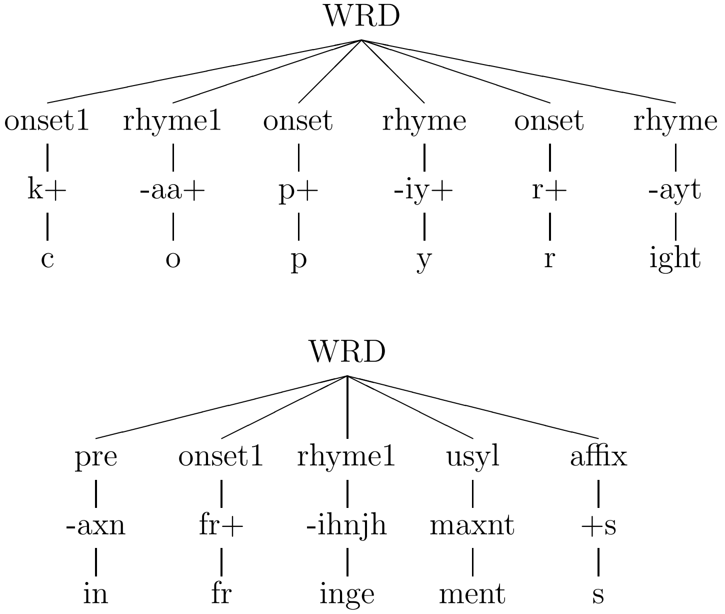


Figure 3-3: Parse tree representation of the phrase *copyright infringements* as defined by the linguistically-motivated context-free grammar. Below the word, the context-free grammar models three hierarchical layers: the sub-syllabic structure, the sub-word (pronunciation) units, and the spelling. In the sub-syllabic layer, the units modeled are: *onset1*, the first stressed onset, *rhyme1*, the first stressed rhyme, *pre*, an unstressed prefix, *usyl*, an unstressed syllable, and *affix*, which models consonants that violate the sonority scale rule in the coda. The nodes in the third layer model the sub-word units, which can be viewed as phoneme clusters with positional markers. $+$ at the end of the sub-word denotes onset, and $-$ at the beginning marks a rhyme. The final layer maps the sub-word units to a graphemic representation, and consists of letter clusters.

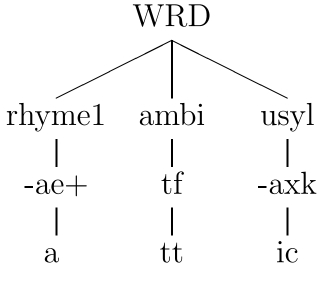


Figure 3-4: A parse tree representation of the word *attic*, illustrating the *ambi* structure. *ambi* is introduced to disambiguate the syllabic assignment of the flapped-/t/.

where [] denotes optional and () denotes OR.

The third layer consists of the pre-terminal nodes and describes all possible ways sub-syllabic categories map to sub-word units. Sub-words can be viewed as intermediate units between phonemes and syllables that encode pronunciation information. Sample rules that illustrate the manner in which various sub-syllabic categories are pronounced are listed below:

```
onset   ⟶ fr+
onset   ⟶ s+
rhyme   ⟶ -ihr
rhyme   ⟶ -ehl
usyl    ⟶ shaxn
usyl    ⟶ maxnt
affix   ⟶ +s
affix   ⟶ +z
```

As illustrated in the samples above, the sub-word units encode positional constraints with a set of diacritics:

<sub-word>+ corresponds to an `onset` unit such as `sh+`.

-<sub-word>+ denotes a `rhyme` that corresponds to a vowel sound such as `-uw+`. In essence, the `rhyme` would only consist of a nucleus.

-<sub-word> denotes one of two cases: (1) a `rhyme` that consists of a vowel sound followed by a consonant such as `-ahn` or (2) a consonant cluster corresponding to an `affix` unit and occurring in the `coda` such as `-st`.

+<sub-word> denotes a consonant cluster corresponding to an `affix` such as `+jh` or the suffixes `+s` and `+z` that could end an `affix` structure

The reader is referred to Appendix A for a description of the grammar. Following the design of the third layer, around 700 sub-word units are generated. Around 480 of the sub-word units are rhymes and 130 are onsets and these account for roughly 480X130 syllables. Previous work in the literature indicates that this number of syllables is sufficient to provide a good coverage of English words. In [Ganapathiraju et al., 2001], 275 syllables covered 80% of the Switchboard database [Godfrey et al., 1992], and the authors chose to model words using only 800 syllables of the 9k that were originally extracted from the data. When the LDC Pronlex English dictionary [Pronlex], which contains around 99k pronunciations, was syllabified in [Bazzi and Glass, 2000b], the result was 14.5k unique syllables. In [Greenberg and Kingsbury, 1997], it is reported that only 12 unique syllables cover 25% of syllables in the English written form, and 339 syllables account for 75%. Moreover, the spoken form exhibited similar characteristics.

The fourth and last layer of the CFG maps the sub-word units to their spellings. Sample rules are illustrated below:

```
-aangk  ⟶ o n ( c | k | x | q | ck )
-aangk  ⟶ a n ( c | ck | k )
+th     ⟶ t h
+th     ⟶ t h e
```

A separately supplied lexicon maps each sub-word unit to its phonemic realization as shown below:

-ayth	ay th
-ehb	eh bd
-uhng	uh ng
yum	y uw m

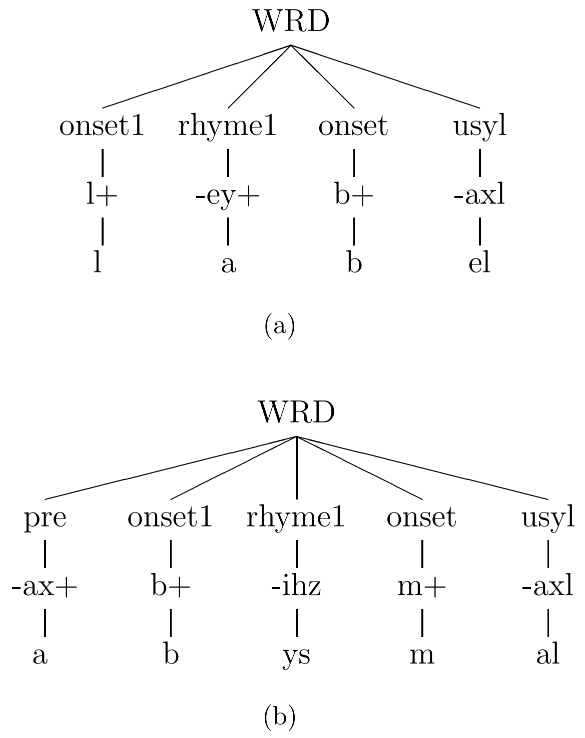
Although our sub-word model stems from linguistic knowledge, pragmatic solutions are taken into consideration while designing the grammar. In our selection of the sub-word units, a trade-off was made between data sparsity and linguistic consistency. For example, as mentioned earlier, a syllable is often split into `onset` and `rhyme`. However, these two categories are combined for a select number of commonly occurring unstressed syllables such as `shaxn` and `maxnt` in order to reduce sparse data and improve probability modeling. Moreover, although the ambisyllabic consonants could be assigned to preceding or following syllables, a separate category is allocated, again to ensure a more compact sub-word lexicon. Hence, in our design, we strove to generate the largest sub-syllabic units that would achieve generality to unseen data yet preserve the compactness of the sub-word lexicon.

The main goal of the CFG is to automatically derive alignments between sub-word units and their spellings (pre-terminals and terminals). Once alignments are derived for a corpus of words, the pre-terminals and terminals associated with each parsed word can be concatenated together, and the result is a set of hybrid units that encode both pronunciation and spelling information. Figure 3-5 illustrates the many-to-many mapping between sound and letter which is encoded in the grammar and which is a typical characteristic of the English language. For example, the sub-word `-axl` can be spelled as either `el` or `al`. Also, the letter `a` can be pronounced `-ax+` or `-ey+`. Following the design of the fourth layer, the total number of hybrid units, which we denote as *spellnemes*, is around 2.5k.

We illustrate below sample sub-word baseforms generated with the grammar:

abatements	-ax+ b+ -eyt maxnt +s
biderman	b+ -ay+ df -er+ maxn
consequential	k+ -axn s+ -ax+ kw+ -ehn sh+ -axl
railcar	r+ -eyl k+ -aar

Next, we illustrate the spellneme representation of the same sample shown above. Each spellneme is of the form `<spelling>_<sub-word>`.



(a)

(b)

Figure 3-5: Parse tree representations of the words *label* and *abysmal* illustrating the many-to-many mapping between sound and spelling in the English language. As demonstrated in the parse trees, the sub-word *-axl* can be spelled as either *el* or *al*. The letter *a* can be pronounced as *-ey+* or *-ax+*. The last two layers in our proposed hierarchical representation are combined to form hybrid units, denoted as spellnemes.

abatements	a_-ax+ b_b+ ate_-eyt ment_maxnt s_+s
consequential	c_k+ on_-axn s_s+ e_-ax+ qu_kw+ en_-ehn ti_sh+ al_-axl
biderman	b_b+ i_-ay+ d_df er_-er+ man_maxn
railcar	r_r+ ail_-eyl c_k+ ar_-aar

As will be shown in Section 3.3, the set of spellnemes will be a key ingredient in the process of designing and training a bi-directional letter-to-sound model.

3.2.3 Previous Work: The Probability Model

The CFG presented in Section 3.2.2 is supported by a probability model, which is trained automatically on data parsed by the grammar. With a hierachical linguistically-motivated framework based on a CFG, it is not immediately apparent how to design a probability model that captures linguistic knowledge from the training data as well as constraints imposed by the grammar. Hence, pragmatic considerations are incorporated in the specifications of the probability model. The context conditions of the probability model are selected to achieve a balance between constraining the data modeling and avoiding sparse data problems. Moreover, unlike stochastic CFGs [Charniak, 1997], the probability model captures conditional statistics on the context of internal parse tree nodes, and not on the production of the associated grammar rule. In particular, probabilities are assigned on sibling-to-sibling (bigram) transitions conditioned on the parent node. The bigram model within each parent category can also be viewed as a trigram model with a spacio-temporal component, which models the probability of each node conditioned on its left sibling and its parent [Seneff, 1992]. The process of training this conditional probability model from parsed data is elucidated through a simple hypothetical example.

Suppose that words in the English language can be modeled by the following two sub-syllabic rules:

```
WRD   ⟶ [onset1] rhyme1 ( ambi | onset ) usyl [affix]
WRD   ⟶ rhyme1 affix
```

The proposed context-free rules are first converted to a network structure by combining rules that share the same left-hand side (LHS) - in this case, the two rules listed above. The network describes all possible interconnections among siblings associated with a particular LHS. `start` and `end` nodes are included as special children of every LHS category to account for the beginning and end of a parse. We illustrate the network structure for the presented example in Figure 3-6.

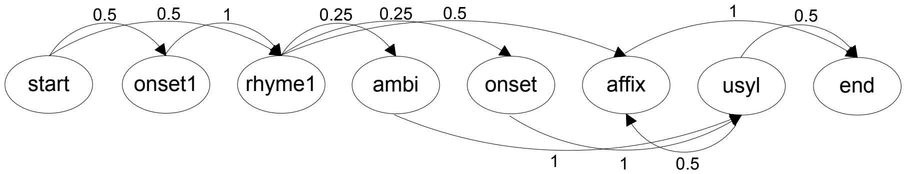


Figure 3-6: The network structure associated with the rules sharing the left-hand side category, WRD. The structure captures the sibling-to-sibling interconnections. Each network arc is weighted by the probability of transitioning to the corresponding right node, given the left sibling and the parent node, WRD. The weights are trained on a corpus of words parsed with the grammar.

Suppose that the following training data and the corresponding parses are provided:

bagels `WRD ⟶ onset1 rhyme1 onset usyl affix`
latin `WRD ⟶ onset1 rhyme1 ambi usyl`
urge `WRD ⟶ rhyme1 affix`
angst `WRD ⟶ rhyme1 affix`

The arc probabilities corresponding to the network structure of each LHS category are trained by counting the number of times a sibling pair associated with a LHS category occurred in the training data and normalizing by the count of the left sibling.

In the case of the network shown in Figure 3-6, if we consider the parsed training data, we note that `onset1` is always followed by `rhyme1`, and this is modeled by the network which has an arc from `onset1` to `rhyme1` with probability 1. On the other hand, a `usyl` ends a word half of the time (*latin*) and is followed by `affix` the other half (*bagels*), and this is also illustrated in Figure 3-6.

Note that the grammar has now generalized to include new rules that were not initially provided. For example, it can now parse the words *rings* and *attics* as follows:

rings	`WRD` ⟶ `onset1 rhyme1 affix`
attics	`WRD` ⟶ `rhyme1 ambi usyl affix`

In addition to modeling the probability of each node conditioned on its left sibling and its parent, some modifications to the probability model are also introduced:

- As illustrated in Figures 3-3 through 3-5 as well as the grammar description in Appendix A, the pre-terminal and terminal nodes in the associated parse trees rarely have left siblings that share the same parent node. If the conditional probabilities of these nodes are computed as discussed earlier, the conditioning would be on the generic `start` symbol, and would capture no context. For this reason, conditioning is done on the parent and the left sibling of a node whether or not that left sibling shares the same parent. The case where the left sibling of a node does not share the same parent is referred to as across-rule training. To avoid inaccurate sparse data modeling, the trigrams, $P(< node > \mid < left\_sibling >, < Parent\_node >)$, obtained during across-rule training are interpolated with the bigram estimates, $P(< node > \mid < Parent\_node >)$. Note that across-rule training is basically conditioning the left-hand side of the grammar rules on external context. Thus the probability model is no longer context-free.

- In order to circumvent sparse data problems at the terminal layer, terminal probabilities are conditioned on the parent of the node and the parent of the left sibling. In essence, the pre-terminals are treated as classes in a class n-gram.

Figure 3-7 illustrates the conditional probabilities that are computed at the terminal, pre-terminal, and sub-syllabic layers. The reader is referred to [Seneff, 1992; Seneff et al., 1992] for further details on the probability model.

Finally, we note that the probability of a unit i, given a preceding unit, j, is the product of the conditional probabilities of all the nodes traversed along the parse tree from j to M and down to i. M is the point where the branches leading to i and j in the parse tree merge.

3.2.4 Previous Work: TINA, The Engineering Framework

TINA, which was introduced in [Seneff, 1992], is a natural language system developed for spoken language applications. The core technology underlying TINA is a context-free grammar defined by hand-written rules as described in Section 3.2.2 and

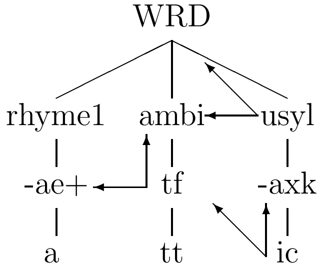


Figure 3-7: A parse tree representation of the word *attic*, illustrating the context conditions for terminals (*ic*), pre-terminals (*tf*), and the sub-syllabic layer (*usyl*). Terminal nodes are conditioned on their parent and the parent of their left sibling. The rest of the nodes are conditioned on their parent and their left sibling irrespective of whether that left sibling shares a parent.

supported by the probability model described in Section 3.2.3. A top-down parsing procedure implemented in a left-to-right fashion uses a best-first search strategy guided by the probability model, which is superimposed on the parse tree defined by the grammar.

In this research, we utilize the engineering principles underlying TINA not for syntactic purposes but in order to design a sub-word model that emulates the English sub-syllabic structure. Hence, instead of training on and parsing sentences, we do so with sequences of sub-syllabic units.

3.3 The Bi-Directional Letter-to-Sound Model

In Section 3.2.1, we stated that the ultimate goal of the linguistic model is to obtain high-quality alignments between graphemic and sound units. These letter-to-sound alignments could then be used to derive a statistical letter-to-sound (L2S) model. The CFG and the supporting probability model, which were presented in Sections 3.2.2 and 3.2.3 respectively, provide us with the mechanism to automatically generate such alignments. By parsing a corpus of words with the CFG framework, we generate alignments between pre-terminal (sub-word) and terminal (spelling) units as observed in Figures 3-3 through 3-5. By concatenating the pre-terminals and terminals of every parse tree, we obtain sequences of spellneme units, which are used to train a spellneme language model. The L2S model presented in this research makes use of a spellneme language model to capture the statistics of spellneme sequences.

The L2S model, $T_{L\rightarrow Ph}$[4], is modeled using finite state transducers (FSTs) [Hetherington, 2004] and is implemented as the composition of four FSTs:

$$T_{L\rightarrow Ph} = T_{L\rightarrow SP} \; o \; G_{SP} \; o \; T_{SP\rightarrow S} \; o \; T_{S\rightarrow Ph} \qquad (3.1)$$

[4] The subscript L→ Ph stands for letter to phoneme.

Where

$T_{L\rightarrow SP}$ is a mapping from letters to spellneme units.

G_{SP} is a spellneme n-gram language model.

$T_{SP\rightarrow S}$ is a mapping from spellneme to sub-word units.

$T_{S\rightarrow PH}$ is a mapping from sub-words to phonemes.

Since the spellneme units consist of concatenations of spelling and sub-word units, it is fairly easy to derive two spellneme lexicons in terms of spelling and sub-word units respectively. A sample spelling lexicon is illustrated below:

```
a_-ax+      : a
ate_-eyt    : a t e
b_b+        : b
ment_maxnt  : m e n t
s_+s        : s
```

A corresponding sub-word lexicon is illustrated below:

```
a_-ax+      : ax
ate_-eyt    : -eyt
b_b+        : b+
ment_maxnt  : maxnt
s_+s        : +s
```

The spelling and sub-word lexicons are used to derive $T_{L\rightarrow SP}$ and $T_{SP\rightarrow S}$ respectively. $T_{S\rightarrow PH}$, on the other hand, is obtained from a separately provided lexicon that maps sub-words to their phonemic representation. The lexicon is provided in Table A.4, Appendix A.

The aforementioned L2S structure can easily be inverted and used as a sound-to-letter (S2L) model. Hence, the proposed framework is used to implement a bi-directional L2S model.

We illustrate the L2S process for the word *abysmal*. When converting from letter to sound, the word *abysmal* is converted to an FST which is essentially a filter that only accepts that word as illustrated in Figure 3-8.

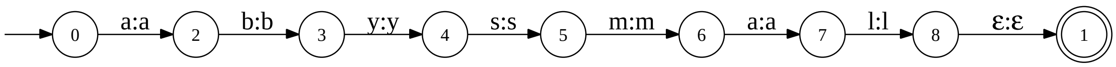


Figure 3-8: A simple finite state transducer representation of the word *abysmal*. Each arc has a label of the form <input>:<output>. ε denotes the null symbol. For example, ε:ε denotes a null transition that does not absorb or emit any symbols. The structure acts as a filter that only accepts the word *abysmal*.

The resulting FST is composed with $T_{L \to SP}$ which maps the letters to a hybrid spellneme representation as illustrated in Figure 3-9.

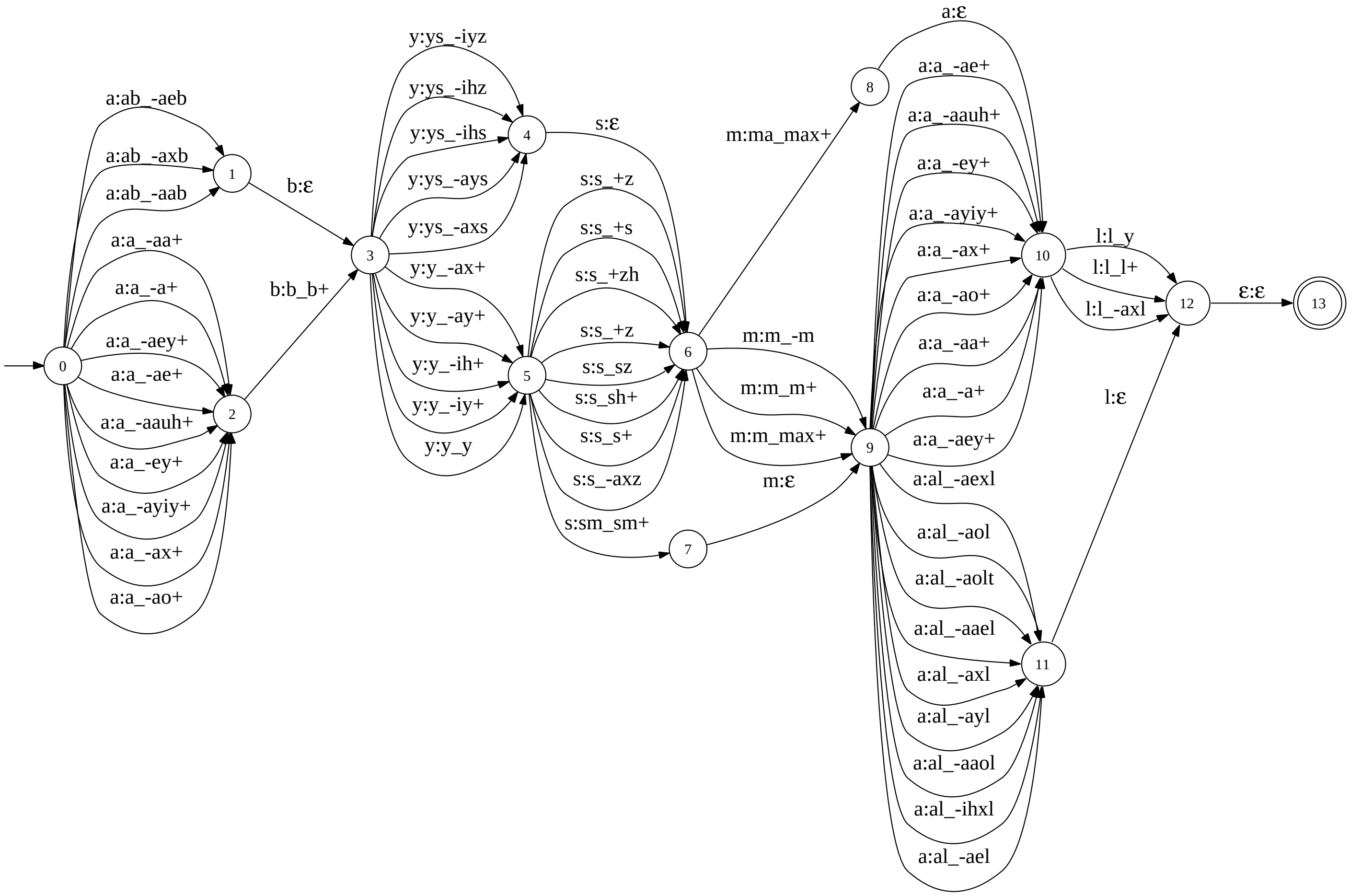


Figure 3-9: A finite state transducer that maps the spelling of the word *abysmal* to a spellneme representation, of the form <spelling>-<sub-word>.

Composing the FST in Figure 3-9 with G_{SP} results in a weighted network that captures the statistics of spellneme strings. Once the weighted network is composed with $T_{SP \to S}$, a letter-to-sub-word model is obtained. In the *abysmal* example, the top sub-word representation obtained following this last composition is "`-ax+ b -ihz -m -axl`".

Another way to view the result incorporated in Figure 3-9 is as a two-stage process:

Letter clustering which amounts to segmenting the word, in this case *abysmal* into all possible letter clusters as illustrated by the sample segmentations below:

```
a b y s m a l
ab ys m a l
ab y s m al
ab y sm al
...
```

Mapping to sub-words which maps the letter clusters to all possible pronunciations as follows:

ab $\longrightarrow$ -aeb
ab $\longrightarrow$ -axb
ab $\longrightarrow$ -aab
ys $\longrightarrow$ -iyz
ys $\longrightarrow$ -ihz
ys $\longrightarrow$ -ihs
ys $\longrightarrow$ -ays
...

Finally, $T_{S \rightarrow Ph}$ maps the sub-words to their phonemic representation, and the top phonemic pronunciation obtained for *abysmal* is "`ax b ih z m ax l`". One can imagine a similar process in the opposite direction in order to achieve sound-to-letter conversion, e.g. generating the word *abysmal* from the pronunciation "`ax b ih z m ax l`".

Hence, a search through $T_{L \rightarrow Ph}/T_{Ph \rightarrow L}$ produces a graph or an *N*-best list of phonemic pronunciations/spellings corresponding to the input spelling/pronunciation.

Chapter 4

Automatic Lexical Pronunciation Generation and Update

This chapter is the first in a series of four that evaluate the linguistically-motivated sub-word units and the bi-directional letter-to-sound (L2S) model in various experimental set-ups. In this chapter, we assess the performance of the L2S model and the sub-word recognizer on the task of automatic lexical pronunciation generation. We define the term "pronunciation" as a sequence of phonemes or a phonemic transcription. Moreover, lexical pronunciation generation is defined as the process of learning and producing the phonemic transcription of a lexicon.

We first describe lexical dictionaries and their role in automatic speech recognizers (ASR). Then, we propose two approaches for automatically generating lexical dictionaries:

1. Using the **letter-to-sound (L2S) model**, which takes letter sequences as input and generates phonemic transcriptions.
2. Using the **sub-word recognizer**, which takes instances of spoken words as input and generates sub-word sequences, which are then converted to phonemic transcriptions.

The research presented in this chapter assumes perfect knowledge of the spelling of the lexicon, and this knowledge is inherently embedded in both of the approaches. In the L2S approach, the spelling of the lexicon is used as input to the system that produces the phonemic transcriptions. In the sub-word recognition approach, the spelling of the lexicon is necessary in order to map the obtained phonemic transcriptions to the appropriate words and create lexical entries. The generated lexical dictionaries are embedded in an ASR system and evaluated in terms of Word Error Rate (WER) on an isolated word recognition task.

4.1 Introduction

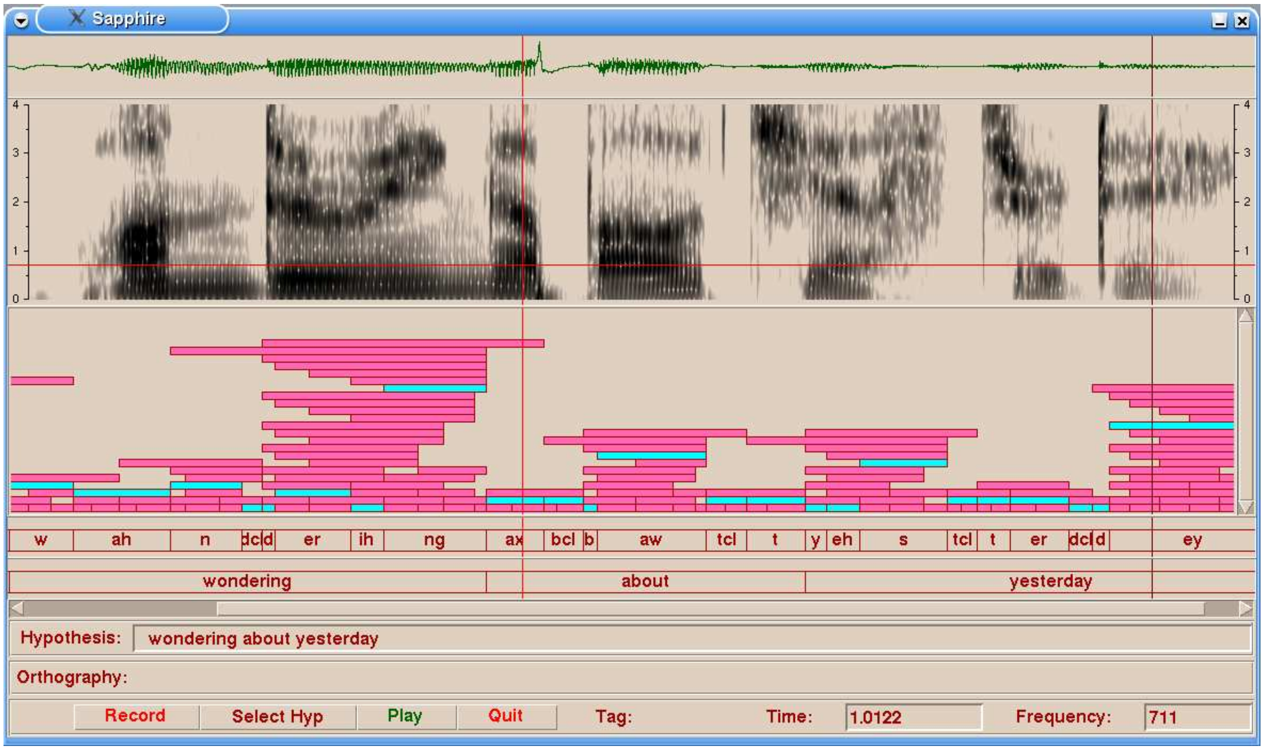
Most automatic speech recognizers (ASR) use a lexical dictionary that maps words to one or more canonical pronunciations. Lexical pronunciations are represented in terms of sub-lexical units which are typically phonemes. Table 4.1 illustrates sample lexical entries, where each entry is a word and its corresponding pronunciation. The pronunciations are transcribed using the ARPABET phonetic alphabet. The reader is referred to Appendix B for further detail regarding the phonetic representation. Figure 4-1 illustrates the incorporation of the dictionary in Table 4.1 in a speech recognizer and the estimation of the word transcription *wondering about yesterday* from a phonetic graph. This particular example makes use of SUMMIT, the landmark-based speech recognizer [Glass, 2003], which is described in more detail in Chapter 2. The example illustrates the use of lexical knowledge to constrain the phonetic graph and output a string of words.

Word	Phonetic Pronunciation
about	ax b aw t
wondering	w ah n d er ih ng
yesterday	y eh s t er d ey

Table 4.1: Sample canonical pronunciations corresponding to the words *about*, *wondering*, and *yesterday*. The pronunciations are transcribed using the ARPABET phonetic alphabet, where the single-letter phones are pronounced like their corresponding English letter. The remaining are pronounced as follows: *[ax]* as in **a**bout, *[aw]* as in l**ou**d, *[ah]* as in m**u**d, *[er]* as in b**ir**d, *[ih]* as in b**i**d, *[ng]* as in si**ng**, *[eh]* as in y**e**s, and *[ey]* as in d**ay**. The reader is referred to Table B.1 in Appendix B for further detail on the phonetic representation.

A dictionary is typically transcribed by lexical experts and is often statically embedded in a speech recognizer. However, when ASR systems are deployed in applications that constantly evolve such as broadcast news transcription, music queries, or restaurant reservation systems, they require constant changes to their dictionaries to account for new words that are often application-specific keywords. One possible solution to this problem is to provide these applications with access to larger dictionaries. However, this solution is not always advantageous. For example, in this chapter, we consider a 2k lexicon of valid restaurant and street names collected for a restaurant reservation domain. Examples of these words are *aceituna, jonquilles, lastorias, pepperoncinis, chungs.* Of these 2k words, 500 are found in a 150k dictionary, 600 words are found in a 300k Google subset, and 1.4k words are found in a 2.5 million Google subset [Google]. Thus, even as larger datasets are considered, a substantial portion (30%) of the 2k lexicon is never found. This is not totally surprising, since the restaurant business is constantly in flux and new restaurants are always emerging. An alternative solution is to routinely and manually update the dictionary. However, this can be time-consuming and prone to error, particularly when the words are unfamiliar or foreign-sounding, such as proper names or restaurants.

In this research, the phonemic dictionary is automatically learned and updated using (1) a letter-to-sound (L2S) model, and (2) spoken instances of words in the lexicon which are presented to a sub-word recognizer. Both approaches are illustrated in Figure 4-2 for the word *abbondanza.* In Figure 4-2(a), the L2S approach is depicted: (a) the word *abbondanza* is first segmented into possible letter clusters, where the



Hypothesis: wondering about yesterday

Orthography:

Tag:

Time: 1.0122

Frequency: 711

Figure 4-1: A graphical interface to the decoding process in the SUMMIT landmark-based speech recognizer [Glass, 2003]. The top 2 panes correspond to the acoustic waveform and its spectrogram. The third pane depicts the network of hypothesized phonetic segments. The best scoring phonetic sequence corresponding to the blue (darker) segments is then shown. This is followed by the corresponding word transcription.

segmentation is dictated by the spellneme units as described in Section 3.3. Next (b) the letter clusters are transformed to sub-word units, which are finally (c) converted to a phonemic representation using the mapping provided in Table A.4, Appendix A. Steps (a), (b), and (c) are performed by the L2S model within the FST framework described in Chapter 3, Section 3.3. The resulting phonemic sequences are concatenated with *abbondanza* to form its lexical entry in the dictionary. In Figure 4-2(b), the sub-word recognition approach is depicted: (a) a spoken instance of the word *abbondanza* is presented to a sub-word recognizer, which outputs an *N*-best list of sub-word sequences. (b) The sub-words are then mapped to a phonemic representation and a lexical entry is generated.

The output of each approach is a lexical dictionary, which is embedded in an ASR and assessed on an isolated word recognition task in terms of Word Error Rate (WER).

The task of automatically generating word pronunciations is not recent, and there has been some research in this domain using decision trees [Bahl et al., 1991] and phonetic decoding [Maison, 2003; Fosler et al., 1996; Sloboda and Waibel, 1996; Westendorf and Jelitto, 1996]. Several researchers have also addressed the problem of

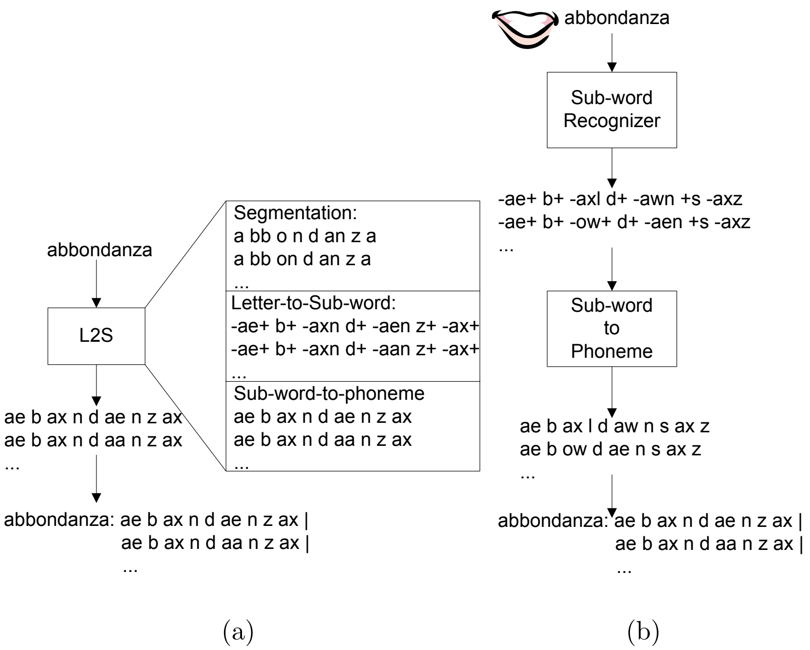


Figure 4-2: An illustration of the two implemented approaches for automatically learning phonemic pronunciations. In Figure 4-2(a), the L2S model takes as input the word *abbondanza*, and generates its phonemic transcription(s). In Figure 4-2(b), a spoken instance of the word *abbondanza* is presented to the sub-word recognizer, and its corresponding phonemic sequence(s) is/are generated.

L2S modeling [Chen, 2003; Bisani and Ney, 2005; Galescu, 2003; Decadt et al., 2002; Chung et al., 2004; Seneff et al., 1996]. This work is different in that it uses the linguistically motivated context-free grammar (CFG) developed in Chapter 3 to design a bi-directional L2S model that is used to learn the *seed pronunciations* of a lexicon. The seed pronunciations are then updated by presenting spoken utterances of words in the lexicon to a sub-word recognizer and using the top N hypotheses as pronunciations. All the generated dictionaries are evaluated on an isolated word recognition task in terms of word error rate. Several experiments described in this chapter are inspired by research conducted by Chung et al. [Chung et al., 2004]. However, the set-ups differ in the L2S model implementation as described in Chapter 3. In addition, we use a larger evaluation data - a 2k restaurant and street name lexicon - in this research as opposed to the 200 names from the OGI corpus used in [Chung et al., 2004].

In this chapter, the following questions are addressed:

1. How good is the quality of a lexical dictionary automatically generated by the L2S model?

2. How good is the quality of a lexical dictionary generated using spoken utterances and the sub-word recognizer?

3. How much improvement is obtained if the spelling of a word is used to constrain the search space of the sub-word recognizer?

4.2 The Implementation Components

In this section, we briefly desribe the L2S model and the sub-word recognizer. For a more detailed overview, we refer the reader to Chapter 3.

The Letter-to-Sound Model ($T_{L \rightarrow Ph}$)

At the core of the L2S model, $T_{L \rightarrow Ph}$, proposed in this research are the spellneme units presented in Chapter 3. The spellnemes form a bridge between letter and sound units and vice versa. The L2S model first segments the spelling of a word into letter clusters, which are mapped to their spellneme counterpart. Spellneme statistics are captured in a spellneme trigram, G_{SP}. The spellnemes are mapped to sub-words, which are then converted to a phonemic representation. Hence, a search through $T_{L \rightarrow Ph}$ produces an N-best list of phonemic pronunciations corresponding to the input spelling. Recall that a sound-to-letter (S2L) model can be similarly implemented.

The Sub-Word Recognizer

The sub-word recognizer is used to automatically generate phonemic transcriptions from spoken utterances of words. Recall that a sub-word recognizer is mathematically modeled as:

$$W^* \approx \underset{S,U,W}{\texttt{argmax}} P(A|S,U,W)P(S|U,W)P(U|W)P(W) \tag{4.1}$$

Where

W^* is the most likely sequence of words.

A is the set of acoustic observations.

S denotes all possible segmentations of the acoustic waveform.

U denotes all possible phone sequences.

$P(A|S,U,W)$ corresponds to a diphone acoustic model.

$P(S|U,W)$ is the duration model, which is kept constant.

$P(U|W)$ is the pronunciation model.

$P(W)$ is the language model.

The reader is referred to Chapter 2, Section 2.1 for details on the derivation of Equation 4.1.

The sub-word search space is implemented within a weighted finite state transducer framework. Decoding is then viewed as finding the most likely path through the composition $O\ o\ R$. O denotes the acoustic-phonetic graph, which models all possible acoustic segmentations. R denotes the pronunciation graph and is itself the composition of four FSTs:

$$R = C\ o\ P\ o\ L\ o\ G \tag{4.2}$$

Where

C denotes the mapping from context-dependent diphone labels to context-independent phone labels.

P represents the phonological rules that map phone labels to phoneme sequences.

L denotes the sub-word lexicon which maps phoneme sequences to sub-word units.

G is the sub-word n-gram language model.

A search through $O\ o\ R$ produces an N-best list of sub-word sequences corresponding to the spoken word. The output of the sub-word recognizer is mapped to a phonemic representation using a sub-word-to-phoneme transducer obtained with the mapping provided in Table A.4, Appendix A.

In some of the experiments in Section 4.4.2, the sub-word recognizer search space is constrained with the spelling of the word. The constraining FST, K, is generated by composing the spelling of a word with the letter-to-sub-word model as illustrated in Figure 4-3. The constraint, K, is then used to generate a spelling-constrained sub-word search space, R_K, as follows:

$$R_K = C\ o\ P\ o\ L\ o\ K\ o\ G \tag{4.3}$$

Hence, a search through R_K produces an N-best list of pronunciations that best match the spelling of the spoken word.

4.3 Data Collection

For the purpose of this research, a list of ~2k restaurant and street names in Massachusetts is selected as the lexicon. Data collection is conducted to record spoken instances of the 2k words. These particular words are of interest to us because they form critical vocabulary in our multimodel restaurant guide domain [Gruenstein and Seneff, 2006]. The names are purposefully chosen to have relatively low Google hit counts as reported by the Google n-gram corpus [Google]. It is worth noting that data collection is conducted for two purposes:

1. To generate phonemic transcriptions from the collected spoken instances using the sub-word recognizer.

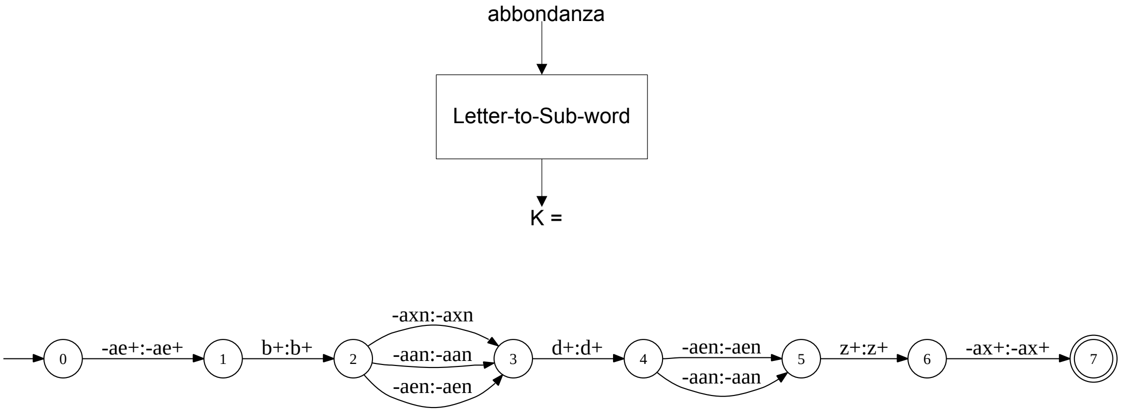


Figure 4-3: The generation of a pronunciation graph for the word *abbondanza* using the letter-to-sub-word module. The pronunciation graph is used to constrain the search space of the sub-word recognizer.

2. To evaluate the generated lexical dictionaries on an isolated word recognition task.

An online user interface is implemented for the purpose of data collection [Choueiter et al., 2007]. The set-up is initially designed within a more general framework to evaluate the sub-word recognizer. First, each subject is presented with a word and is prompted to speak it. A sub-word recognizer complemented with an S2L model is used to generate hypothesized spellings of the spoken word. The spellings are then filtered using the 2k lexicon, and the top 5 candidates are presented to the subject. If the correct spelling is not in the proposed list, the subject is prompted to speak the word again. The same process is then repeated, and a new list of top 5 candidates is presented to the subject. If, again, the correct spelling is not in the proposed list, the subject spells the word. Hence, the sub-word recognizer is given two chances to get the correct word, after which, a letter recognizer is activated. The end result is that each word in the lexicon is recorded at least once, a subset is recorded twice, and a smaller subset is recorded twice along with a spelling. The data collection process just described is illustrated in Figure 4-4. The data collected in spelling mode is integrated into an unsupervised algorithm for automatic lexical dictionary generation, which is described in Chapter 5.

Excluding the data recorded in spelling mode, 2842 utterances are collected from 19 speakers - 12 males and 7 females - and the spoken utterances pertaining to each word are recorded by the same speaker. A breakdown and description of the collected data is shown in Table 4.2. As implied by Table 4.2, the lexicon of Set2a and Set2b is one and the same.

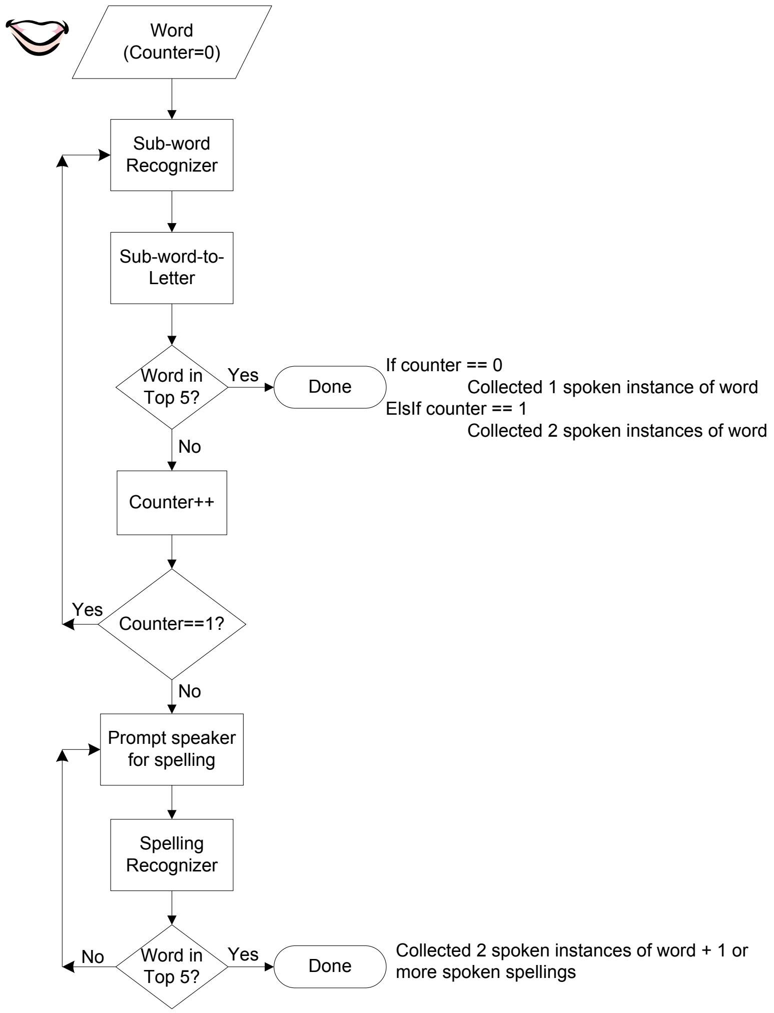


Figure 4-4: Flowchart depicting the data collection process for the restaurant and street names. Subjects are presented with a name and are prompted to speak it. The sub-word recognizer has two chances to get the correct hypothesis, after which the subjects are asked to spell the word.

4.4 Experiments

In all the experiments in this chapter, the SUMMIT landmark-based speech recognition system is used [Glass, 2003]. The spellneme trigram, G_{SP}, is trained on a 300k-word subset of the Google corpus [Google]. The Google corpus originally contains ∼13 million unique words, and is very noisy. It is reduced to ∼2.5 million words by only keeping lower-cased words with alphabetic symbols. The corpus is then intersected with a carefully cleaned ∼500k lexicon and is augmented with nouns from the Phonebook development set and Pronlex. The result is a ∼300k clean corpus of commonly used English words.

The sub-word n-gram, G, is a trigram trained on sub-word representations of the 300k Google words obtained with the L2S model. Finally, the isolated word recognizer

Name	Size	Description
Set1	1142	Instance of words spoken once
Set2a	850	First instance of words spoken twice
Set2b	850	Second instance of words spoken twice

Table 4.2: Description of the collected data. A total of 2842 utterances are obtained for a 2k lexicon. Set2a and Set2b share the same lexicon and correspond to the list of words recorded twice during data collection.

has a 2k vocabulary as described in Section 4.3, and uses a word unigram.

Section 4.2 describes the automatic generation of the phonemic pronunciations using the L2S model and reports on results. Section 4.4.2 describes the pronunciation update process which uses spoken instances of the lexicon, and the sub-word recognizer. Results are reported for pronunciations generated with the unconstrained as well as spelling-constrained sub-word recognizer. Section 4.4.3 reports the results obtained when the pronunciations generated by the different setups are combined.

4.4.1 Pronunciations Generated with the L2S Model

In this section, we report the results obtained for the phonemic pronunciations automatically generated with the L2S model. First, the 2k lexicon is presented to the L2S model and the $\text{top}_n \mid n = 1, ..5, 10, 20, 50$ pronunciations are generated for each word. As the aforementioned description implies, the L2S approach assumes perfect knowledge of the spelling of the lexicon. We illustrate below the top 2 L2S pronunciations for three sample words:

```
yainnis  : ( y ay n ax s | y ey n ax s )
squantum : ( s k w aa n t ax m | s k w aa n tf ax m )
shawarma : ( sh ao aa r m ax | sh ax w aa r m ax )
```

After generating a dictionary for the 2k lexicon, an isolated 2k-word recognizer is built and evaluated on Set1, Set2a, and Set2b. The results are reported in terms of WER in Table 4.3. We first observe that Set1 has a lower word error rate (WER) than Set2a and Set2b. This is expected since Set1 is the set of words that are recognized in the first round during data collection and is likely, therefore, to be an *easier* set than Set2a and Set2b. Next, the WER of Set2b is lower than that of Set2a. One possible explanation is that subjects tend to speak the words more carefully in the second round upon failing the first one. Finally, as expected, the WER improves significantly as the number of alternative pronunciations is initially increased. For example, compared to the top 1 pronunciation results, the top 10 results exhibit an absolute improvement of 9.2%, 4.9%, and 7.8% on Set1, Set2a, and Set2b respectively. The WER starts deteriorating as pronunciation confusion is increased, in this case beyond 20 pronunciations.

For comparison purposes and to evaluate the effectiveness of the L2S model at generating lexical pronunciations, manual corrections are carefully introduced by a

	Set1	Set2a	Set2b
top 1	25.7	52.4	47.8
top 2	20.3	47.9	42.8
top 3	17.9	47.6	41.2
top 4	17.3	47.3	39.9
top 5	17.1	47.8	39.5
top 10	16.5	47.5	40.0
top 20	16.9	48.5	40.2
top 50	18.6	47.8	42.6

Table 4.3: WERs of the 2k-word recognizer on the three data sets, Set1, Set2a, Set2b as a function of the top$_n$ $\mid n = 1, ..., 5, 10, 20, 50$ pronunciations generated by the L2S model.

lexical expert into the top 1 pronunciations obtained with the L2S model. As shown in Table 4.4, absolute improvements of 2.2%, 1.9%, and 3.1% are obtained for Set1, Set2a, and Set2b respectively. The modest improvements observed following manual corrections is encouraging since it indicates that the L2S model is very good at generating valid pronunciations. In fact, in comparing Table 4.3 with Table 4.4, it is noted that a system that includes just two automatically produced alternative pronunciations outperforms a system that utilizes a single manual pronunciation for each lexical entry.

	Set1	Set2a	Set2b
Original top 1	25.7	52.4	47.8
Manually corrected top 1	23.5	50.5	44.7

Table 4.4: WERs of the 2k-word recognizer before and after the phonemic dictionary generated by the L2S model is manually corrected. The results are reported for the top 1 phonemic pronunciations on the three data sets, Set1, Set2a, Set2b.

4.4.2 Pronunciations Generated with the Sub-Word Recognizer

We proceed, in this section, to report the results for the pronunciations generated with the sub-word recognizer described in Section 4.2. First, we recall that Set2a and Set2b correspond to the first and second spoken utterances of the same set of words. The words in Set2a are presented to the sub-word recognizer and the generated top$_n$ $\mid n = 1, 2..5$ sub-word sequences are converted to phonemic transcriptions using the mapping provided in Table A.4, Appendix A. The phonemic sequences replace those generated by the L2S model in Set2b. On the other hand, since there is only one recorded instance of the words in Set1, the phonemic transcriptions corresponding to Set1 are still generated by the L2S model. The pronunciations are concatenated to their corresponding words in the lexicon to form lexical entries in the dictionary.

This mapping requires the knowledge of the spelling of the lexicon, which is an underlying assumption in this chapter. The sample top 2 pronunciations obtained with the sub-word recognizer are illustrated below:

```
yainnis  : ( y uw n ax s | y uw n ax eh s td )
squantum : ( s w ih n s ah m  | s w ih tq en )
shawarma : ( sh w ao r m | sh w ao r m l ax s )
```

Following this procedure, an updated 2k phonemic dictionary is obtained, and a new 2k-word recognizer is built. Since, in this section, the pronunciations are learned from Set2a, the recognizer is evaluated only on Set1 and Set2b.

It is first noted that the Set2b pronunciations generated by the sub-word recognizer perform better than those obtained with the L2S model. For example, the top 5 WER of Set2b improves by an absolute 2.2% (39.5% to 37.3%). It can be deduced that the lexical dictionary generated for Set2b by the sub-word recognizer is a more suitable representation than the one obtained with the L2S model. This is possibly because the sub-word based pronunciations are generated from Set2a. The reader is reminded that Set2a and Set2b consist of the first and second spoken instances of the same set of words, and the spoken utterances corresponding to each word are recorded by the same speaker. In other words, the performed recognition task is speaker-dependent and the sub-word recognizer is capable of capturing speaker characteristics well, whereas the L2S model cannot since it does not make use of spoken data.

The results of Set1 exhibit a different trend than those of Set2b. Although the pronunciations of Set1 are still generated by the L2S model, the top$_n$ WERs of Set1 shown in Table 4.5 are consistently worse than those observed in Table 4.3. For example, the top 5 WER of Set1 deteriorates by an absolute 2% (17.1% to 19.1%). One possible explanation is that the lexicon of Set2b is well modeled by the sub-word recognizer to the extent that the resulting pronunciations of Set2b are competing with those of Set1.

	Set1	Set2b
top 1	27.8	45.9
top 2	23.4	42.0
top 3	20.1	39.8
top 4	19.4	37.8
top 5	19.1	37.3

Table 4.5: WERs of the 2k-word recognizer evaluated on Set1 and Set2b as a function of the top$_n$ | $n = 1, 2..5$ pronunciations generated by the sub-word recognizer. The pronunciations of Set1 are still generated by the L2S model.

Next, the spelling of each word in Set2a is presented to the L2S model and a corresponding spelling-constrained lattice, K, is generated. The resulting top$_n$ | $n = 1, 2..5$ pronunciations replace those previously generated by the L2S model. Similarly as before, the lexical pronunciations of Set1 are still generated by the L2S model. As illustrated below, the top 2 pronunciations obtained with the constrained sub-word recognizer are closer to the canonical pronunciations than the ones obtained with the unconstrained model.

```
yainnis  : ( y ey n ax s | y ay n ax s )
squantum : ( s k w aa n t ax m | s k w aa n td ax m )
shawarma : ( sh ax  w ao r m ax  | sh ao w ao r m ax )
```

Table 4.6 illustrates the WERs of Set1 and Set2b as a function of the top$_n$ | $n = 1, 2..5$ pronunciations. Compared to the L2S pronunciations, the top 1 WER for Set1 has an absolute deterioration of 0.8%, which is substantially better than the 2.1% deterioration obtained with the unconstrained sub-word pronunciations. On the other hand, the top 1 absolute improvement for Set2b has dramatically increased from 1.9% to 12.2%.

	Set1	Set2b
top 1	26.5	35.6
top 2	22.1	34.0
top 3	19.8	33.6
top 4	19.3	32.4
top 5	19.1	32.7

Table 4.6: WERs of the 2k-word recognizer evaluated on Set1 and Set2b as a function of the top$_n$ | $n = 1, 2..5$ pronunciations generated by the spelling-constrained sub-word recognizer for words spoken twice.

4.4.3 Pronunciations Combination

So far, we have replaced the L2S pronunciations of the words in Set2a with the ones acquired from the spoken utterances. We now proceed to combine the different acquired pronunciations and report on WERs in Tables 4.7, 4.8, and 4.9. It is important to note, again, that whereas the Set2b lexicon has alternative pronunciations obtained from the spoken utterances, the Set1 lexicon does not. For example, if #total pronunciations = 4 and # sub-word pronunciations = 2, this implies that, for the Set2b lexicon, the last two L2S pronunciations are replaced with those obtained with the sub-word recognizer. On the other hand, for the Set1 lexicon, all 4 pronunciations are from the L2S model.

Table 4.7 shows the WERs of Set1 and Set2b as a function of both total number of pronunciations as well as number of pronunciations generated with the sub-word recognizer. The observed trend is for the WERs of Set1 and Set2b to decrease as the total number of pronunciations is increased. However, for a fixed total number of pronunciations, the performance of Set1 suffers while that of Set2b improves, as more L2S pronunciations are replaced with sub-word pronunciations. This trend is consistent with the previously observed results in Tables 4.3 and 4.5 where the WER improves as the number of alternative pronunciations is initially increased. Furthermore, the increased pronunciation confusion introduced by the spoken utterances leads to performance deterioration for Set1.

# total pronunciations	# sub-word pronunciations	Set1	Set2b
2	1	22.2	33.8
3	1	20.1	32.2
3	2	20.8	32.8
4	1	19.7	29.9
4	2	20.1	31.2
4	3	20.1	31.2
5	1	18.8	31.6
5	2	19.3	30.6
5	3	19.7	29.9
5	4	19.9	29.9

Table 4.7: WERs of the 2k-word recognizer evaluated on Set1 and Set2b as a function of combined pronunciations. The first column is the total number of pronunciations, and the second column is the number of sub-word pronunciations for words spoken twice.

Table 4.8 exhibits similar behaviour as Table 4.7 except that the sub-word pronunciations are generated with a spelling-constrained sub-word search space. We observe that combining the spelling-constrained pronunciations with the L2S pronunciations does not result in as much gain as that reported in Table 4.7. One possible explanation is that the spelling-constrained pronunciations are not very different from the L2S pronunciations, and hence do not introduce as much *new information* to the L2S pronunciations as the unconstrained sub-word pronunciations.

Finally, Table 4.9 reports the best results for Set2b, which are obtained when the L2S pronunciations are combined with those generated by both unconstrained and spelling-constrained sub-word recognizers.

4.5 Summary and Discussion

In this research, we have presented a new approach towards the automatic learning of lexical pronunciations. We have evaluated our approach on an isolated word recognition task for a 2k lexicon of restaurant and street names.

The linguistically-motivated CFG-based L2S model is used to learn the seed pronunciations of the lexicon. To assess the performance of the L2S model, the top 1 L2S pronunciations are manually corrected and evaluated. The modest improvement obtained with the manual modifications indicates the effectiveness of the L2S model. The lexical pronunciations are then refined using spoken utterances of the lexicon, which are presented to a sub-word recognizer. Our best results are obtained when the L2S pronunciations are combined with both spelling-constrained and unconstrained

# total pronunciations	# constrained sub-word pronunciations	Set1	Set2b
2	1	20.8	34.4
3	1	18.1	34.0
3	2	19.3	33.3
4	1	17.9	35.3
4	2	18.9	34.0
4	3	19.0	32.2
5	1	17.9	35.2
5	2	18.5	35.1
5	3	19.2	33.6
5	4	19.1	32.2

Table 4.8: WERs of the 2k-word recognizer on Set1 and Set2b as a function of combined pronunciations. The first column is the total number of pronunciations, and the second column is the number of spelling-constrained sub-word pronunciations for words spoken twice.

# total pronunciations	# L2S pronunciations	# sub-word pronunciations	# constrained sub-word pronunciations	Set1	Set2b
3	1	1	1	20.3	29.8
5	1	2	2	20.9	27.9

Table 4.9: WERs of the 2k-word recognizer on Set1 and Set2b as a function of combined pronunciations. The first column is the total number of pronunciations, the second, third, and fourth columns are the number of L2S, unconstrained, and spelling-constrained sub-word pronunciations for words spoken twice.

sub-word pronunciations. To provide easy comparisons among the different experiments, we show in Table 4.10 the results for several experiments where the total number of pronunciations is held constant at 3. For Set1, the best result is with the L2S pronunciations, and the least deterioration (0.2% absolute) is obtained when the L2S pronunciations are combined with the constrained sub-word pronunciations. For Set2b, the constrained sub-word pronunciations perform better than the unconstrained setup as well as the L2S pronunciations. Furthermore, combining the three types of pronunciations provides the best results for Set2b and the best overall results.

Although we implement and evaluate our model on an isolated word recognition task, we envision our approach implemented in open-ended continuous-speech applications. For example, given audio waveforms and their corresponding word transcription, the L2S model and sub-word recognizer can be used to automatically update the dictionary corresponding to the data. Other applications involve open-ended spoken queries that allow users to introduce manual corrections in case of transcription errors. Both spoken utterances and corrections can be used to update the lexical baseform of a pre-existing word or add the baseform of a new word to the dictionary.

Table	4.3	4.5	4.6	4.7	4.8	4.9
# L2S pronunciations	3	0	0	2	2	1
# sub-word pronunciations	0	3	0	1	0	1
# constrained pronunciations	0	0	3	0	1	1
Set1 WER	17.9	20.1	19.8	20.1	18.1	20.3
Set2b WER	41.2	39.8	33.6	32.2	34.0	29.8

Table 4.10: Comparison of the WERs of Set1 and Set2b as a function of pronunciations. The first row refers to the Table number of the original experiment. The second, third, and fourth rows are the number of L2S, sub-word, and constrained sub-word pronunciations respectively.

In this research, we have assumed perfect knowledge of the spelling of a word. In other scenarios, such as spoken dialogue systems, the user might provide a *spoken* rendering of the spelling of a word. In Chapter 5, we propose and implement an unsupervised iterative algorithm in which spoken instances of a word and its spelling are used to learn lexical pronunciations.

Chapter 5

Turbo-Style Algorithm: An Unsupervised Approach Towards Lexical Dictionary Estimation

In Chapter 4, we proposed and implemented an approach towards automatically learning lexical pronunciations using the letter-to-sound (L2S) model as well as spoken instances of words which are presented to a sub-word recognizer. In the previously proposed method, we assumed perfect knowledge of the spelling of words in the lexicon.

In this chapter, we pursue further the task of automatic lexical acquisition, and relax the assumption of perfect spelling knowledge. We propose an iterative and unsupervised algorithm, denoted Turbo-style, which presents spoken instances of both spellings and words to a letter and sub-word recognizer respectively, and fuses information from both systems to boost the overall lexical learning performance. The algorithm is used to automatically learn the phonemic dictionary of the restaurant and street names lexicon described in Section 4.3, Chapter 4, and is evaluated in terms of spelling accuracy, letter error rate (LER), and phonetic error rate (PER) of the lexical entries. The automatically generated lexical dictionaries are also evaluated on an isolated word recognition task in terms of word error rate (WER).

5.1 Introduction

The process of learning or updating the lexical dictionary of an ASR system can be triggered by newly acquired information such as a spoken instance of a word or its spelling. In the previous chapter, efforts were concentrated on learning a lexical dictionary using only spoken renderings of a set of words. In this chapter, spoken instances of the spellings of the words are also taken into consideration when learning the lexical entries in the dictionary.

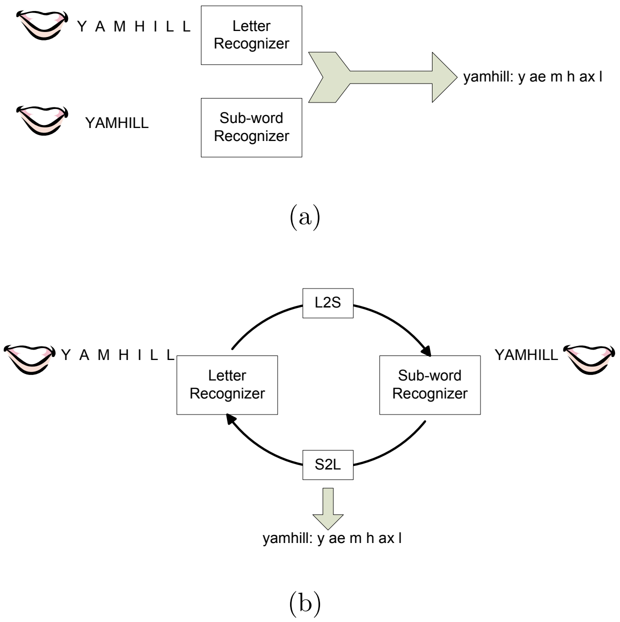
The ability to automatically learn a reliable estimate of a lexical entry (both spelling and phonemic transcription) of a word from spoken examples, can prove quite beneficial. For example, consider spoken dialogue systems, which have been emerging as a natural solution for information retrieval applications [Zue et al., 2000]. Such systems often suffer from dialogue breakdown at critical points that convey crucial information such as named entities or geographical locations. One successful approach proposed for error recovery in dialogue systems lies in speak-and-spell models, that prompt the user for the spelling of an unrecognized word [Schramm et al., 2000; Filisko and Seneff, 2005]. Figure 5-1 illustrates an example of such an error recovery mechanism in a flight reservation domain where the user is attempting to reserve a flight to *Yamhill.* In such a case, *Yamhill* is not in the dictionary of the flight reservation domain, but a spoken rendering of the spelling of the word as well as the word itself have been provided by the user. The question that this research attempts to answer is: Given spoken instances of both the spelling and the word, how well can a valid lexical entry in a dictionary be learned?

U	I need a flight from Riga, Michigan to Yamhill, Oregon on May ninth.
S	From Riga, Michigan. Please spell the name of your arrival city?
U	Y A M H I L L.

Figure 5-1: Sample dialogue from a flight reservation domain where the user, U, is trying to reserve a flight to the city *Yamhill* that the system, S, does not *know.*

Given spoken instances of a word as well as its spelling, a straightforward approach is to present each to a sub-word and letter recognizer respectively, and to select the top-1 outputs in order to generate a lexical entry. This approach is illustrated for the word *Yamhill* in Figure 5-2(a). However the research presented in this chapter improves upon this simple method by introducing an unsupervised iterative technique denoted *Turbo-style algorithm.* A simple depiction of the Turbo-style algorithm is illustrated in Figure 5-2(b), where spoken instances from two complementary domains - spelling and pronunciation - are presented to a letter and sub-word recognizer respectively. The output of each recognizer is then processed by a bi-directional L2S model and injected back into the other recognizer in the form of *soft* bias information. Such a set-up is denoted Turbo-style learning algorithm since it is inspired by the principles of Turbo Codes [Berrou et al., 1993]. The term Turbo Code is in turn a reference to turbo-charged engines where part of the output power is fed back to the engine to improve the performance of the whole system.

The novel contribution of this work is two-fold: (1) spoken examples of both the spelling and the word are used as opposed to the word only, and (2) a bi-directional L2S model is used to exchange bias information between the spelling and pronunciation domain to boost the overall performance of the tandem model. It is worth noting that the set-up does not consult a lexicon when estimating the spelling.



(a)

(b)

Figure 5-2: Illustrations of two possible approaches towards learning a lexical entry given spoken renderings of a word as well as its spelling. A straightforward method is depicted in Figure 5-2(a), with the word and its spelling presented to a sub-word and letter recognizer respectively and the top 1 hypotheses selected. The Turbo-style algorithm is illustrated in Figure 5-2(b), where, instead of just selecting the top 1 hypotheses, the recognizers are allowed to exchange bias information through the bi-directional L2S model.

5.2 The Turbo-Style Algorithm

In this section, the Turbo-style iterative algorithm is presented. The basic principle behind the proposed algorithm is to have two complementary recognizers, letter and sub-word, exchange bias information such that the performance of both systems is improved. In this particular implementation, the letter recognizer first generates an N-best list, which is projected into the sub-word domain using an L2S model. The projected N-best list is used to bias the sub-word LM, by injecting into it the pronunciations that best match the estimated spelling. A mirror procedure is performed in the sub-word domain. The algorithm is illustrated in Figure 5-3, and the steps for a pair of spoken spelling and word are as follows:

1. The spoken spelling is presented to the letter recognizer, and a letter N_1-best list is generated.

2. The letter N_1-best list is transformed to a sub-word M_1-best list using the L2S model.

3. A bias sub-word language model (LM) is trained with the sub-word M_1-best

list, and interpolated with a base sub-word LM by a factor w_1. The interpolated LM becomes the new base sub-word LM.

4. A sub-word recognizer is built with the new interpolated sub-word LM.

5. The spoken word is presented to the sub-word recognizer, and a sub-word M_2-best list is generated.

6. The sub-word M_2-best list is processed by the S2L model, and a letter N_2-best list is produced.

7. A bias letter LM is trained with the letter N_2-best list, and is interpolated with a base letter LM by a factor w_2. The interpolated LM becomes the new base letter LM.

8. A letter recognizer is built with the new interpolated letter LM.

9. Go back to Step (1).

The aforementioned description of the Turbo-Style algorithm as well as Figure 5-3 show that 7 parameters need to be tuned:

1. N_1: the size of the spelling N-best list generated by the letter recognizer.

2. M_1: the size of the sub-word N-best list produced from the spelling N_1-best list using the L2S model.

3. w_1: the weight assigned to the bias sub-word LM.

4. M_2: the size of the sub-word N-best list generated by the sub-word recognizer.

5. N_2: the size of the spelling N-best list produced from the sub-word M_2-best list using the S2L model.

6. w_2: the weight assigned to the bias letter LM.

7. K, the total number of times the algorithm is iterated.

The tuning of these parameters is described in Section 5.4.

5.3 Experimental Set-Up

The spellneme trigram, G_{SP} used by the L2S/S2L model is built with 300k parsed nouns extracted from the Google corpus [Google]. The letter trigram, G_L, is also trained with the 300k Google words, and the sub-word trigram, G_S, with the same set converted into sub-words using the L2S model.

For the purpose of this research, 603 Massachusetts restaurant and street names were recorded together with their spoken spellings. This set is part of a larger data collection effort described in more detail in [Choueiter et al., 2007], and in Section 4.3, Chapter 4. The 603 spelling/word pairs are split into a development (Dev) set of 300 pairs used to tune the Turbo algorithm and a Test set of 303.

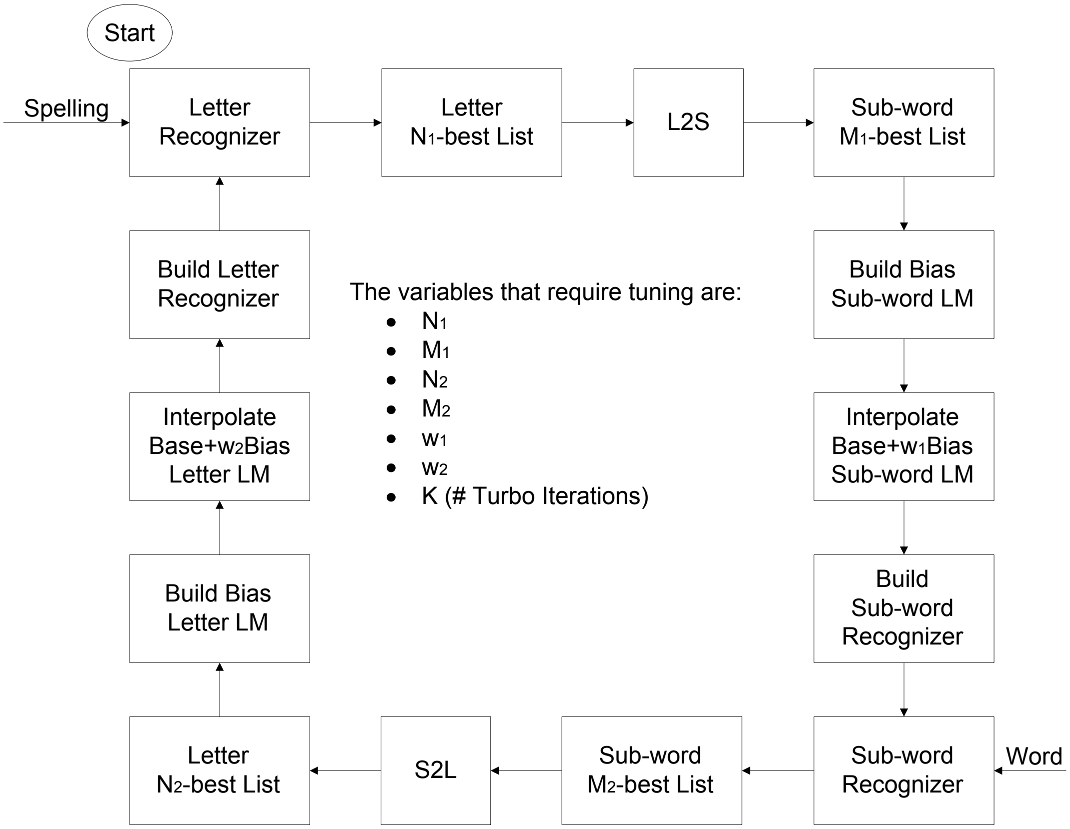


Figure 5-3: Illustration of the iterative and unsupervised Turbo-style algorithm used to refine the estimates of the spelling and the pronunciation of a new word. The algorithm presents spoken instances of a word and its spelling to a sub-word and letter recognizer respectively. The recognizers then bias each others' LMs with their respective N-best outputs. The N-best outputs are projected from one domain to the other using a bi-directional L2S model.

5.4 Parameter Tuning

In this section, the process of setting the parameters of the algorithm is presented. There are various ways of approaching this problem, and the choice here is to set N_1 and M_2 separately, while M_1 and w_1 are tuned simultaneously, and similarly for N_2 and w_2. Furthermore, each parameter is tuned by considering particular modules of the Turbo-style algorithm separately. It is worth emphasizing that an empirical approach is adopted for parameter tuning, and the author makes no claim of optimality of the approach.

$\mathbf{N_1}$ and $\mathbf{M_2}$ correspond to the number of top candidate spellings and pronunciations generated by the letter and sub-word recognizers respectively. N_1 is chosen to achieve an effective compromise between capturing the correct spelling and weeding out incorrect ones. This is done by presenting the Dev data to the letter recognizer and monitoring the depth of the correct spelling in the top 100 candidates. By this process, N_1 is empirically set to 20.

In a similar procedure on the pronunciation side, M_2 is empirically set to 50. However, it is worth noting that, while reference spellings are available for the letter set-up, no references are available for the sub-word set-up. To avoid having to manually transcribe sub-word baseforms, the L2S model is used to automatically generate them similarly to the approach taken in Chapter 4.

$\mathbf{N}_2$ **and** $\mathbf{w}_2$ denote the number of top candidate spellings produced by the S2L model and the weight of the biased letter LM respectively. The two parameters control the amount of bias injected into the letter LM, and are tuned to improve the performance of the letter recognizer on the Dev set. Performance is evaluated in terms of *spelling match rate*. A spelling match occurs when the correct word is in the N_1-best list generated by the letter recognizer, where $N_1 = 20$. Since $M_2 = 50$, a sub-word 50-best list is processed by the S2L, producing a spelling N_2-best list, where $N_2 = 20$, 100, 500, 1000, 5000, 10000. For each value of N_2, a biased LM is trained with the spelling N_2-best list and interpolated with a base LM. The interpolation weight, w_2 is varied between 0 and 1 in 0.2 steps. For each (N_2,w_2) pair, a letter recognizer is built and the spelling 20-best list is generated. Figure 5-4 reports the performance in terms of spelling match rate as a function of N_2 and w_2, and illustrates that mid-range values of both N_2 and w_2 are best. For example, the performance deteriorates when either too much or too little weight is given to the biased LM. Based on this, N_2 is set to 1000 and w_2 to 0.4.

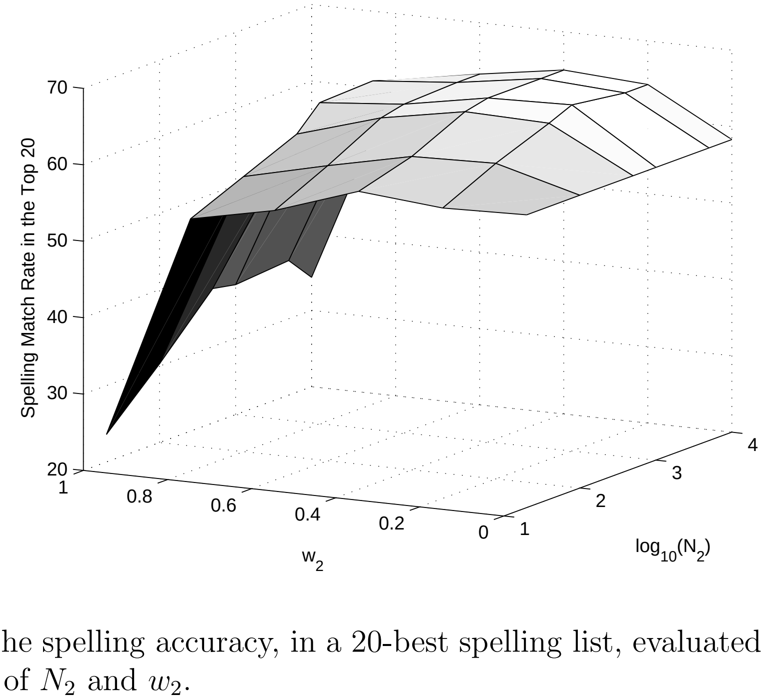


Figure 5-4: The spelling accuracy, in a 20-best spelling list, evaluated on the Dev set as a function of N_2 and w_2.

$\mathbf{M}_1$ **and** $\mathbf{w}_1$ correspond to the number of top candidate sub-word sequences generated by the S2L model and the weight of the biased sub-word LM respectively. They are tuned similarly to N_2 and w_2, and M_1 is set to 1000 and w_1 to 0.8. The weight assigned to the biased sub-word LM is smaller than the one assigned to the biased letter LM ($\mathrm{w}_1 < \mathrm{w}_2$) indicating that the sub-word recognizer is more confident about the bias information obtained from the letter domain than vice versa. This is possibly because the spelling domain with its smaller alphabet is more constrained and hence more reliable than the sub-word one.

K corresponds to the number of iterations of the Turbo-style algorithm. To set K, the algorithm is run on the Dev set until little change in performance is observed. Performance is measured in terms of top N spelling and pronunciation match rates, as well as top 1 letter and phonetic error rate in the lexical entry. The top N match rate reflects the number of times the correct answer is found in the top N hypotheses. The results are reported in Tables 5.1-5.4, where the first column is always the algorithm iteration number. Iteration 0 refers to the initial results of the letter and sub-word recognizers prior to receiving any feedback information from the complementary domain, as illustrated in Figure 5-2(a). In Tables 5.1-5.2, the second to fifth columns show the spelling and pronunciation match rates in the top 1, 10, 20, and 100 spelling and pronunciation candidates respectively. The reader is reminded that manually transcribed reference pronunciations are not available. Hence, for the purpose of reporting results in the pronunciation domain, reference pronunciations are generated with the L2S model, and are then manually edited by a lexical expert.

Turbo	Spelling Match Rate			
Iteration #	Top 1	Top 10	Top 20	Top 100
0	18.7%	50.6%	57.6%	77.6%
1	24.3%	53.6%	62.3%	78%
2	25%	56.3%	62.6%	76.6%
3	25%	56%	62.6%	76.6%

Table 5.1: Top 1, 10, 20, and 100 spelling match rates on the Dev set as the Turbo-style algorithm is iterated 3 times. The top N match rates indicate the frequency at which the correct spelling is found in the top N candidates.

Turbo	Pronunciation Match Rate			
Iteration #	Top 1	Top 10	Top 20	Top 100
0	0%	0.66%	1.33%	3.66%
1	3.66%	13%	18.33%	32.33%
2	4.33%	12.66%	21%	34.33%
3	4%	13.66%	20.33%	34.33%

Table 5.2: Top 1, 10, 20, and 100 pronunciation match rates on the Dev set as a function of algorithm iterations.

The results in Tables 5.1-5.2 show substantial improvement in the spelling and pronunciation match rates following iteration 2. For example, the top 1 spelling and pronunciation accuracies improve by an absolute 6.3% and 4% respectively. While it is surprising to see very low accuracies for the sub-word recognizer (0% top 1 accuracy at the 0^{th} iteration), it is important to note that a hypothesized sub-word sequence might still be valid even though it does not match the reference pronunciation. This is due to the fact that words can have multiple pronunciations, particularly a lexicon of restaurants and street names. This is further illustrated by examples in Table 5.3. Moreover, unlike the spelling references, there is no guarantee that the pronunciation generated with the L2S model and then manually edited are indeed *true* references.

Word	Reference Pronunciation	Hypothesized Pronunciation
hewitts	`hh yu+ -axt +s`	`hh yu+ -iht +s`
gallos	`g+ -ael -ow+ +z`	`g+ -ehl -ows`
anadolu	`-axn -ae+ df -axl -uw+`	`-aen -ax+ d+ -axl -uw+`

Table 5.3: Sample words from the restaurant lexicon with their corresponding reference sub-word based pronunciations generated by the L2S model and the hypothesized pronunciation proposed by the sub-word recognizer. The sample results suggest that words can have multiple valid pronunciations.

Table 5.4 reports the letter and phonetic error rates (LER and PER) on the Dev set as a function of Turbo-style algorithm iterations. Similarly to the match rate results, a significant improvement is observed following only 2 iterations. The LER and PER exhibit absolute improvements of 4.8% and 19.1% respectively.

Iteration #	Top 1 LER	Top 1 PER
0	25.3%	63.2%
1	21.1%	43.8%
2	20.5%	44.1%
3	20.6%	44.3%

Table 5.4: Top 1 letter and phonetic error rates on the Dev set as a function of algorithm iterations.

Based on the results in Tables 5.1-5.4, and the observation that no significant improvement occurs beyond iteration 3, K is set to 2.

5.5 Results and Discussion

The parameters are adjusted based on the Dev set as described in Section 5.4 such that $(N_1, M_2, N_2, w_2, M_1, w_1, K) = (20, 50, 1000, 0.4, 1000, 0.8, 2)$, and preliminary results are obtained on the Test set.

a In Section 5.5.1, the quality of the generated lexical entries is evaluated as a function of the Turbo-style algorithm iterations. Each lexical entry in the dictionary consists of a word and its pronunciation. Hence, lexical entries are assessed in terms of spelling and pronunciation match rate as well as letter and phonetic error rate.

b In Section 5.5.2, the generated dictionaries are embedded in an isolated word recognizer, and evaluated on spoken instances of the restaurant and streetname lexicon in terms of word error rate.

5.5.1 Accuracies and Error Rates of the Lexical Entries

Similar to the results reported on the Dev set, Tables 5.5-5.6 show significant improvement in the spelling and pronunciation match rates of the lexical entries. For example, the top 1 spelling and pronunciation accuracies improve by absolute 7.2% and 5.3% respectively following 2 iterations. The letter error rate reported in Table 5.7 is also found to decrease from 22.8% in iteration 0 to 19.1% in iteration 2 (16.2% relative improvement).

The algorithm also substantially improves the almost-correct spelling rate. In this case, almost-correct spelling is when the edit distance between the top 1 spelling and the correct one is no more than 1 letter. The almost-correct rate increases from 43.2% at iteration 0 to 52.8% at iteration 2. This suggests that a spelling correction has a better chance of finding the reference word in a lexicon retrieved, say from the World Wide Web.

	Spelling Match Rate			
Iteration #	Top 1	Top 10	Top 20	Top 100
0	20.5%	54.1%	66.3%	77.2%
1	26.4%	57.8%	66.9%	80.2%
2	27.7%	59.1%	66.9%	79.2%

Table 5.5: Top 1, 10, 20, and 100 spelling match rates on the Test set as a function of iterations.

	Pronunciation Match Rate			
Iteration #	Top 1	Top 10	Top 20	Top 100
0	0.33%	0.33%	0.66%	2.33%
1	3.96%	16.83%	21.12%	33.99%
2	5.61%	16.5%	20.79%	35.64%

Table 5.6: Top 1, 10, 20, and 100 pronunciation match rates on the Test set as a function of iterations.

Iteration #	Top 1 LER	Top 1 PER
0	22.8%	62.8%
1	19.7%	43.1%
2	19.1%	43.1%

Table 5.7: Top 1 letter and phonetic error rates on the Test set as a function of iterations.

Table 5.8 illustrates qualitative improvements in the pronunciation of sample words from iteration 0 to iteration 2. It also demonstrates the point made in Section 5.4, where a valid hypothesized pronunciation might not be a perfect match to the corresponding reference. For example the final pronunciation of the word *olivio* is valid but does not match the reference `-ow+ l+ -ihv -iy+ -ow+`.

Word	Iteration 0	Iteration 2
botoloph	`-ao+ tf -ow+ l+ -aof`	`b+ -owt -axl -aolf`
quans	`-eyn +z`	`kw+ -aan +z`
olivio	`l+ -ey+ df -iy+ -ow+`	`-axl -iy+ v+ -iy+ -ow+`
woodmans	`-ahn m+ -aen s+ -ihng`	`w+ -uhd m+ -aen +s`
churrascaria	`jh+ -ehs t+ -ehr -iy+ -ax+`	`ch+ -aoer+ -axs k+ -ehr -iy+ -ax+`

Table 5.8: Sample pronunciations (in sub-word units) generated by the Turbo-style algorithm at iterations 0 and 2. The results show significant qualitative improvement in the pronunciations following the use of the feedback mechanism in the algorithm.

Similarly, Table 5.9 illustrates sample words and their corresponding spelling improvement from iteration 0 to iteration 2. As shown in Table 5.9, the bias information obtained from the pronunciation domain could drive the spelling recognizer to a local optimum which does not match the reference, e.g. *tartufo*, and vice versa. Hence, the optimality of the proposed scheme remains to be examined. For example, instead of keeping the parameters N_1, M_2, N_2, w_2, M_1, and w_1 static, it might be more advantageous to adaptively update them to reflect the confidence in the bias information.

Word	Iteration 0	Iteration 2
mcmenamy	mcnenanys	mcmenamys
tartufo	cruso	cartufo
terranova	trialve	trianove
helmand	heelmand	helmand
scutra	setra	scutra

Table 5.9: Sample spellings produced by the Turbo-style algorithm at iterations 0 and 2. Two out of five of the examples exhibit a full recovery following 2 iterations. The word *tartufo* has an almost-correct recovery, and *terranova* a partial recovery.

5.5.2 Isolated Word Recognition Results

At each iteration of the Turbo-style algorithm, the top 1 hypotheses of the letter and sub-word recognizers are concatenated to form a lexical entry in a dictionary. The learned lexical entries are *imperfect* in the sense that either the spelling of a word or its pronunciation or both could be faulty. Table 5.10 illustrates a portion of the learned dictionaries following each iteration of the algorithm.

Reference Dictionary	
botoloph : b ao tf ax l ao f	
woodmans : w uh d m ax n z	
helmand : h eh l m ax n dd	
Generated Phonemic Dictionary	
Iteration 0	Iteration 2
botollpah : ao tf ow l ao f	botollph : b ow td ax l ao l f
wordmans : ah n m ae n s ih ng	wordmans: w uh dd m ae n s
heelmand : hh aw m ax n td	helmand : hh eh l m eh n td

Table 5.10: A portion of the phonemic dictionary learned by the Turbo-style algorithm. The top portion corresponds to the reference lexical entries generated by the L2S model. The first and second columns in the second portion correspond to the entries generated by the Turbo algorithm in iterations 0 and 2.

The *imperfect* phonemic dictionaries are then each used to implement isolated word recognizers and are evaluated in terms of Word Error Rate (WER). The performances of the recognizers are compared to that of an isolated word recognizer built with a reference dictionary, which is, in turn, generated by the L2S model and manually edited by a lexical expert. This process is illustrated in Figure 5-5.

The evaluation data consists of spoken instances of the words in the Test set. The reader is reminded that the Test set was used to generate the lexical dictionaries whereas the evaluation data will be used to assess the generated dictionaries on an isolated word recognition task. Though both sets share the same lexicon, they consist of different spoken instances. Table 5.11 reports the WERs of the word recognizers implemented with both reference and Turbo-generated dictionaries. Following two iterations of the Turbo algorithm, the recognizer built with the final Turbo-generated dictionary has a WER of 20.8%, and exhibits a dramatic relative decrease in WER of 63.5% compared to the first Turbo-generated dictionary. Although the reference dictionary is originally superior to the Turbo-generated dictionary obtained at iteration 0, it is immediately outperformed after one Turbo iteration. Following two Turbo iterations, the recognizer associated with the final Turbo-generated dictionary has a 42.7% relative improvement in WER over the reference dictionary. This improvement is likely due to the fact that the learned lexical entries approximate the actual pronunciations of the users more closely than the canonical forms.

5.6 Summary and Discussion

In this research, an iterative and unsupervised Turbo-style algorithm is introduced and implemented for automatic lexical learning. A spoken example of a word and its spelling are presented to a sub-word and letter recognizer, which recursively exchange bias information through a bi-directional L2S model. As a proof of concept, preliminary experiments are conducted using 603 pairs of spoken spellings and words, where half of the set is used for development and the rest for testing.

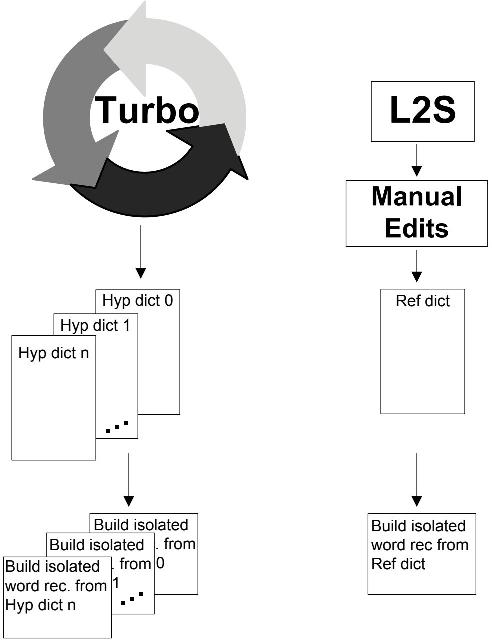

Turbo
L2S
Manual Edits
Hyp dict 0
Hyp dict 1
Hyp dict n
Ref dict
Build isolated word rec. from 0
Build isolated word rec. from 1
Build isolated word rec. from Hyp dict n
Build isolated word rec from Ref dict

Figure 5-5: Illustrations of the phonemic dictionaries learned using the Turbo algorithm, and the reference dictionary generated using the L2S model followed by manual editing. The dictionaries are then used to build isolated word recognizers.

The quality of the generated lexical entries is evaluated in two manners:

1. The spelling accuracy and the letter error rate of the generated lexical entries exhibited significant absolute improvements of 7.2% and 3.7% respectively following two iterations of the Turbo algorithm. The pronunciation accuracy and the phonetic error rate of the learned pronunciations also showed similar trends with absolute improvements of 5.28% and 19.7% respectively.

2. The phonemic dictionaries learned at each iteration of the Turbo algorithm are embedded in isolated word recognizers and evaluated in terms of word error rate. The WER improved by an absolute 13.2% following two iterations of the algorithm.

Within the same Turbo framework, it remains important to investigate (1) different schemes for parameter tuning, (2) other methods for exchanging bias information between different domains, as well as (3) extensions of this algorithm to more general set-ups. As future work, the algorithm is also expected to be incorporated into a spoken dialogue system for automatically acquiring new words.

Finally, the basic principle of the proposed algorithm is the fusion of several sources of information, and it can be generalized to different set-ups. For example, a recent approach to unsupervised pattern discovery in speech produces reliable clusters of similar speech patterns [Park and Glass, 2008]. The generated clusters can be processed by multiple sub-word recognizers whose outputs can be fused to boost the pronunciation recognition performance.

Iteration #	WER
0	34.0%
1	23.1%
2	20.8%
Reference WER	
29.7%	

Table 5.11: The word error rates of the isolated word recognizers built with the learned (*imperfect*) phonemic dictionaries. The WER of the recognizer built with the reference dictionary is also reported. The recognizers are evaluated on 303 isolated words that share the same lexicon as the Test set.

Chapter 6

A Hybrid Approach Towards Open-Ended Recognition Using Sub-Word Modeling

Chapters 4 and 5 addressed the problem of automatic lexical learning. In this and the following chapter, the focus shifts towards improving word recognition through sub-word modeling. In particular, in this chapter, we propose and implement a preliminary evaluation of the sub-word units in the context of isolated word recognition. Specifically, a sub-word recognizer is embedded in a parallel fashion as a back-off mechanism for a word recognizer. The resulting hybrid model is evaluated in a lexical access application where a user speaks a word and the word recognizer first hypothesizes and displays the top candidate words. If the correct word is not in the returned list, the system backs off to a sub-word recognizer.

6.1 Introduction

One of the factors impeding the broad acceptance of ASR is the frustration experienced by users when the system breaks down when an unknown word occurs. For word recognizers with fixed vocabularies, this problem is inevitable since the recognizer does not have immediate access to the lexical entries corresponding to unknown words. In this chapter, we address the unknown word problem by complementing an isolated word recognizer with an error recovery mechanism based on a sub-word recognizer. The parallel hybrid model - word and sub-word recognizers - is evaluated in a simple lexical access application where a user speaks a word, and an isolated word recognizer (Stage I) proposes and displays a list of top candidate words. If the person rejects all the words, the system enters the second stage (Stage II), which uses the sub-word recognizer. This process is illustrated in Figure 6-1.

The sub-word recognizer generates hypothesized sub-word sequences which are then transformed to word spellings via a sub-word-to-letter mapping that encodes the conditional probability $P(\texttt{letter sequence | sub-word})$. Invalid spellings are filtered through a look-up in a large lexicon. The hybrid model is evaluated on

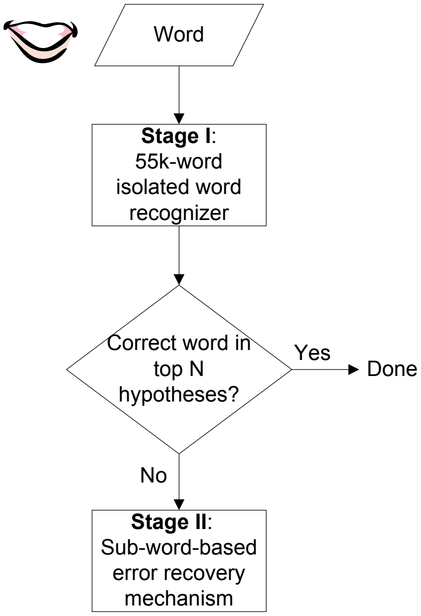


Figure 6-1: A flowchart of the hybrid model, which consists of a 55k-word recognizer complemented with an error recovery mechanism. The back-off mechanism is based on a sub-word recognizer.

4.7k nouns drawn from the Phonebook development set. In order to properly assess the sub-word based error recovery mechanism in Stage II, the evaluation data are purposefully selected to have a reasonably high OOV rate of 31% with respect to the isolated word recognizer in Stage I.

Since the hybrid model has, potentially, an open-ended vocabulary, it is important to compare it to a large-vocabulary isolated-word recognizer. Hence, before evaluating the hybrid model, a 300k-word recognizer is built and assessed on the evaluation data. Since a manually transcribed dictionary corresponding to all of the 300k lexicon is unavailable, it is automatically generated using the L2S model proposed in this thesis. Hence, in the process of building the 300k-word recognizer, we re-evaluate the ability of the L2S model to automatically learn and generate phonemic pronunciations. In Chapter 4, this aspect of the L2S model was evaluated much more rigorously.

In this chapter, we are interested in addressing the following questions: (1) How does the open-ended hybrid system compare with a large-vocabulary isolated word recognizer ? (2) How does the sub-word based error recovery mechanism affect the performance of the isolated 55k-word recognizer ?

6.2 Stage I: The Word Recognizer

Stage I consists of an isolated word recognizer with a 55k-word lexicon drawn from the LDC Pronlex dictionary [Pronlex]. The recognizer is implemented within the SUMMIT framework [Glass, 2003]. Since the task is isolated word recognition, the recognizer is guided by a uniform unigram language model.

6.3 Stage II: The Sub-Word Based Back-Off Mechanism

Stage II consists of a sub-word recognizer which acts as a back-off mechanism to the word recognizer in Stage I. If Stage II is triggered, the sub-word recognizer produces a string of sub-words, which is converted to a graphemic representation through a sub-word-to-letter mapping. The sub-word and spelling estimations can be modeled mathematically as follows:

Given acoustic observations, A, the optimal letter spelling, $L^\star$, can be written as:

$$\begin{aligned} L^\star &= \underset{L}{argmax}\ P(L|A) = \underset{L}{argmax} \sum_S P(L,S|A) \\ &\approx \underset{L}{argmax}\ \underset{S}{max}\ P(L,S|A) \\ &\approx \underset{L}{argmax}\ \underset{S}{max}\ P(A|S)P(L|S)P(S) \end{aligned} \tag{6.1}$$

Where L is a sequence of letters, and S corresponds to the sub-words units. $P(A|S)$ is the acoustic model, $P(S)$ is modeled as an n-gram on the sub-words, and $P(L|S)$ is the conditional probability of a letter sequence given a sub-word sequence. The last line assumes that the acoustic events, A, are conditionally independent of the letters, L, given the sub-words, S, i.e. $P(A|S,L) = P(A|S)$.

The product $P(A|S)P(S)$ models the sub-word search space, which can be implemented as a weighted FST, R [Hetherington, 2004]:

$$R = C\ o\ P\ o\ Lex\ o\ G \tag{6.2}$$

Where C denotes the mapping from context-dependent model labels to context-independent phone labels, P the phonological rules that map phone labels to phoneme sequences, Lex the sub-word lexicon, which is a mapping from sub-word to phonemic units , and G the sub-word language model (LM). A search through R produces an N-best list of sub-word sequences, which is denoted R_{N-best}.

The spelling search space, as represented in Equation 6.1, can be modeled as:

$$L = R_{N-best}\ o\ M_{S2L}\ o\ D \tag{6.3}$$

Where M_{S2L} is a statistical sub-word-to-letter mapping which encodes $P(L|S)$. D is a deterministic word filter or acceptor, and is used to enforce hard spell-checking, such that if the generated spelling is not in some large lexicon, it is rejected. Following the filtering stage, a spelling cohort is generated.

In the rest of this chapter, we refer to the output of R as a sub-word N-best list and the output of L as a spellings cohort.

An illustration of the sub-word based error recovery mechanism in Stage II is shown in Figure 6-2. When an utterance is presented to the sub-word model, an N-best list of sub-word sequences with corresponding acoustic and LM scores is produced by the sub-word recognizer. The acoustic score is combined with a weighted LM score to form a total score for each sub-word sequence. The sub-word list is transformed into an exhaustive spellings cohort by using M_{S2L}, and invalid words are filtered out with D. D, is built with a ∼300k lexicon, which is a subset of the Google corpus [Google]. To model M_{S2L}, the 300k lexicon described in Section 6.4 is first decoded into spellnemes using the parsing mechanism described in Chapter 3. The ML estimate of the conditional probability $P(L|S)$ encoded in M_{S2L} is then obtained simply using counts over the parsed lexicon.

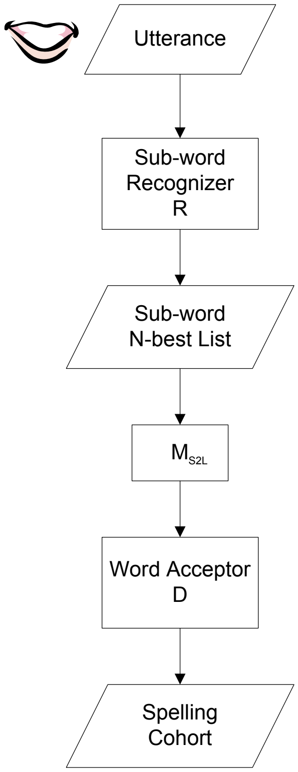


Figure 6-2: A flowchart of the sub-word based error recovery mechanism. The estimation of the final spelling cohort is done by converting the sub-word sequences hypothesized by the sub-word recognizer into spellings using M_{S2L} and filtering the result with the word acceptor, D.

6.4 Evaluation Data

Our evaluations are performed on 4682 nouns drawn from the development set of the Phonebook telephone-quality isolated words corpus [Pitrelli et al., 1995]. The lexicon for the isolated word recognizer in Stage I of the hybrid system consists of 55k nouns extracted from the LDC Pronlex dictionary [Pronlex]. In our experiments, we refer to the Phonebook nouns that are in the 55k lexicon as IV_{55k} (in-vocabulary), and to the words that are not as OOV_{55k}. There are 3228 IV_{55k} and 1454 OOV_{55k} words in the Phonebook nouns.

	Top 10 accuracy			Top 20 accuracy		
	IV$_{55k}$	OOV$_{55k}$	All	IV$_{55k}$	OOV$_{55k}$	All
55k	83%	0%	57%	86%	0%	59%
300k	72%	72%	72%	77%	77%	77%

Table 6.1: Comparison of the 55k and 300k isolated word recognizers, in terms of IV$_{55k}$, OOV$_{55k}$, and overall accuracy. Both recognizers are evaluated based on the top ten and twenty word candidates.

6.5 Experiments and Results

This section describes several experiments conducted on the Phonebook data. First, a large-vocabulary isolated word recognizer is built and assessed on the ∼4.7k-word evaluation dataset. This initial step allows the comparison of the hybrid system to a large-vocabulary recognizer. Prior to evaluating the hybrid system, the sub-word based error recovery mechanism is assessed as a function of the sub-word language model (LM) and the size of the sub-word N-best list.

6.5.1 Large-Vocabulary Isolated Word Recognizer

In order to build a 300k-word recognizer, the L2S model is first used to automatically generate the phonemic pronunciations of the 300k lexicon. It is noted that a subset of this lexicon consisting of Pronlex [Pronlex] and Phonebook nouns already has manually transcribed pronunciations, and these are kept. Pronunciations are automatically generated for the rest of the words.

The 300k-word recognizer is then evaluated on the 4682 Phonebook nouns in terms of top 10 and top 20 accuracies, meaning that success occurs if the correct word is in the top 10 and top 20 candidates respectively. The results are reported in Table 6.1 for the 3228 IV$_{55k}$ and the 1454 OOV$_{55k}$ words separately. For comparison, the 55k-word recognizer is also built and evaluated on the same data sets. We note here that all the evaluated words including the OOV$_{55k}$ words are in the 300k lexicon. As reflected by the results, the performance of the IV$_{55k}$ and OOV$_{55k}$ subsets is the same for the 300k system. This illustrates that the automatically generated pronunciations are performing comparably to the manually transcribed ones. Furthermore, the IV$_{55k}$ words suffer significant degradation with the 300k system compared to the 55k-word recognizer (i.e. 86% to 77% for top 20 accuracy) due to the larger vocabulary.

6.5.2 Sub-Word Language Models

After evaluating the large-vocabulary recognizer in the previous section, we turn to the evaluation of the sub-word based error recovery mechanism in Stage II of the hybrid model.

The sub-word recognizer in Stage II produces an N-best list of sub-word sequences, guided by a sub-word trigram LM, $P(S)$, that is trained on a large corpus. A critical issue is the quality of this LM. In this section, we assess the performance of the

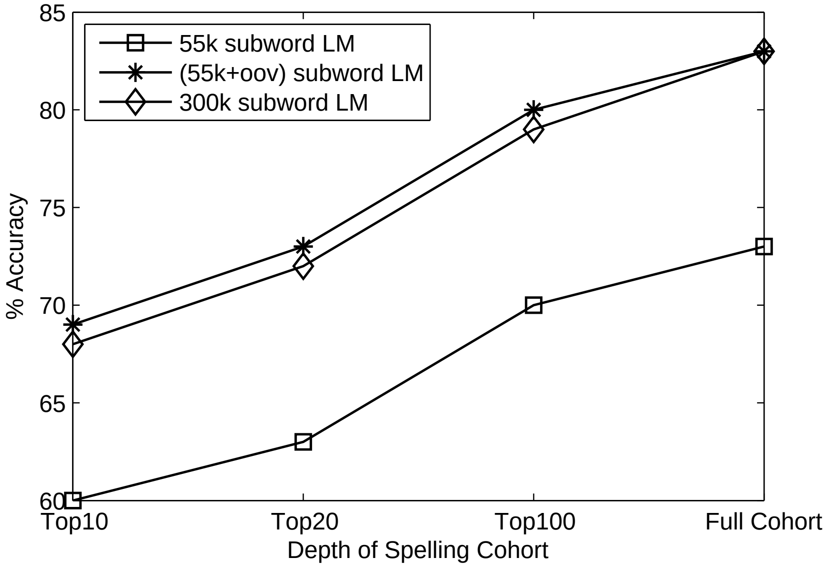


Figure 6-3: Accuracy of the three sub-word recognizers for different depths of the spelling cohort evaluated on the 1454 OOV_{55k} words. The spellings are generated with a sub-word 1000-best list.

sub-word recognizer as a function of several sub-word language models. We train the sub-word LMs from three training corpora parsed into sub-words: (1) the 55k lexicon, (2) the 55k lexicon augmented with just the OOV_{55k} words in Phonebook, and (3) the 300k lexicon. Figure 6-3 compares the performance of the three sub-word recognizers on the OOV_{55k} words. Each of the recognizers produces 1000-best sub-word lists which are then converted into a cohort of all possible valid spellings. A match occurs if the correct word is in the spelling cohort, and we report accuracies on cohorts of sizes 10, 20, and 100, as well as on the whole spelling cohorts. As illustrated in Figure 6-3, the inclusion of only the OOV_{55k} words in the sub-word LM training data results in a substantial improvement in performance (i.e. 60% to 69% for top 10 accuracy). Only a slight degradation is incurred with the full 300k lexicon (i.e. 69% to 68% for top 10 accuracy).

6.5.3 Sub-Word *N*-best Depth

Next, the performance of Stage II is evaluated as a function of sub-word *N*-best depth. The sub-word recognizer is generated with an LM trained on the 300k lexicon. Since the computational requirements of the sub-word model can be significantly reduced with a smaller sub-word *N*-best list, it is of interest to measure degradation in performance as a function of sub-word *N*-best depth, *N*. As illustrated in Figure 6-4, modest degradation is incurred in the top 10 accuracy as *N* is decreased from 1000 to 100 (69% to 66%).

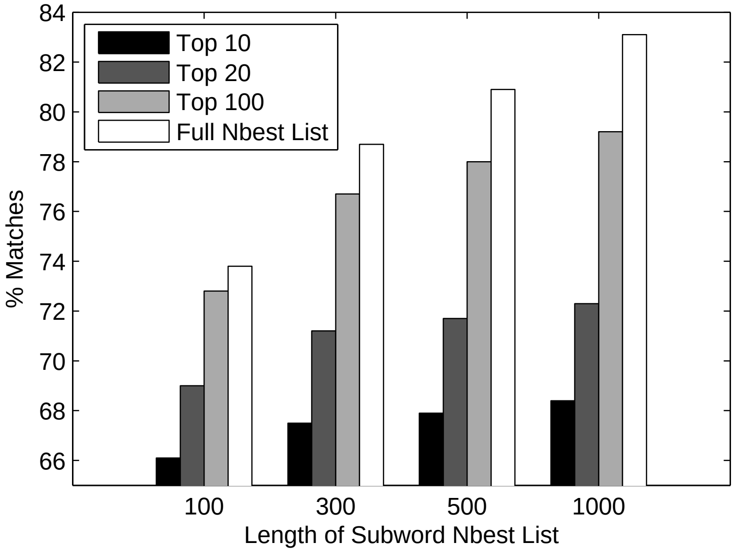


Figure 6-4: The sub-word model accuracy as a function of the depth of the *N*-best list. Accuracy is reported on spelling cohorts of size 10, 20, and 100, as well as on the full spelling cohort. The 300k LM sub-word recognizer is used.

6.5.4 Hybrid System Evaluation

In this section, we evaluate the hybrid system in a lexical access application where a user speaks a word and the 55k-word recognizer generates a 10-best list of candidate words. If the correct word is not in the 10-best list, the sub-word model is triggered, and a spelling cohort of size 10 is generated.

The 4682 Phonebook nouns are used to simulate words spoken by users. The 55k-word recognizer is used in Stage I, and all words that fail to appear in the 10-best list are passed to the sub-word model in Stage II. In this research we focus on the estimation of the spelling of an OOV word, not on the detection of an OOV word. Thus, we rely on direct user feedback to achieve perfect OOV detection. In our experiments, this is simulated by automatically passing all the words that failed Stage I to Stage II.

In this section, we evaluate the overall performance of the multi-stage recognizer for IV_{55k} and OOV_{55k} words. The 55k-word recognizer is used in Stage I, and the 300k LM sub-word recognizer with a 1000-best list of sub-words is used in Stage II. The pie charts in Figure 6-5 describe the percentage of matching words in a spelling cohort of size ten for the word and sub-word recognition stages.words, Stage I proposes the correct word among the top 10 word candidates 83% of the time. If the correct word is not in the top 10, the system reverts to the sub-word model in Stage II. Stage II recovers an additional 1% of the IV_{55k} words, which now make the top-10 cut due to the availability of alternative pronunciations beyond the ones supplied in the lexicon. The top 10 accuracy of Stage II on the OOV_{55k} words is 69%. We note that the top 10 list of Stage II excludes any results from Stage I. Hence, we can compare the overall accuracy of Stages I and II to the top 20 accuracy of the 300k isolated word recognizer

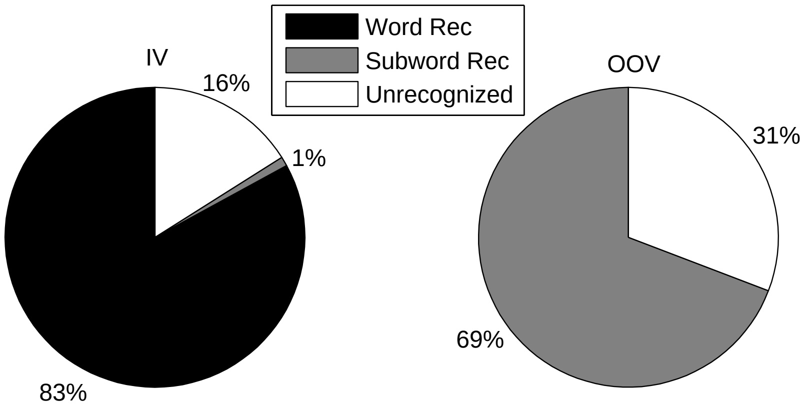


Figure 6-5: Accuracy of the word and sub-word recognition stages for a spelling cohort of size ten evaluated on IV_{55k} and OOV_{55k} words.

shown in Table 6.1. The overall accuracy of Stages I-II is 79%, which outperforms the top 20 accuracy of the 300k isolated word recognizer (77%), most probably due to the more focused 55k-word recognizer in Stage I.

6.6 Summary and Discussion

In this chapter, we incorporated a sub-word recognizer in an error recovery mechanism for an isolated 55k-word recognizer. Preliminary results are reported on ∼4.7k nouns drawn from the Phonebook development set. The back-off mechanism, which used a sub-word recognizer, is evaluated as a function of sub-word LM as well as depth of the sub-word N-best list. The hybrid model is also compared with a more traditional isolated 300k-word recognizer. In the process of building the 300k-word recognizer, the L2S model is used to automatically generate the phonemic pronunciations of the 300k lexicon. It is important to note that the hybrid model described in this chapter consisted of word and sub-word recognizers connected in parallel. In Chapter 7, we propose to improve word recognition performance by implementing a serial configuration of word and sub-word recognizers.

In general one can envision the sub-word model implemented within a dialogue system, thereby taking advantage of user interactions and augmenting the system with a learning capability. The sub-word model would be activated upon the detection of an OOV word, and any newly acquired word could then be added to the lexical dictionary.

Chapter 7

Recognition and Information Retrieval Experiments in the Lyrics Domain

In the previous chapters, the sub-word units were assessed in isolated word recognition set-ups. In Chapters 4 and 5, the sub-word and L2S models were used to generate lexical entries from recorded instances of isolated words - as well as their spoken spellings in Chapter 5. In Chapter 6, a sub-word model was integrated in parallel with an isolated word recognizer, and was manually triggered when an out-of-vocabulary (OOV) word was encountered.

In this chapter, the sub-syllabic sub-word units are embedded in a flat hybrid OOV model for a continuous ASR. We denote the ASR complemented with the sub-word based flat OOV model as hybrid ASR. As opposed to Chapter 6 where a parallel OOV model was implemented, in this chapter, a serial OOV model is explored. The hybrid ASR is deployed as a front-end to a song retrieval application which is queried via spoken lyrics. Using the hybrid ASR, the spoken lyrics are first decoded into one or more strings of words. The recognition output is then converted to an appropriate query representation, which is used to search the song database. The retrieval system is assessed in terms of recognition as well as song retrieval performance.

7.1 Introduction

In this chapter, a set-up is proposed and implemented to evaluate the performance of the sub-word units and the L2S model in a realistic continuous speech task. The main objective is to query a song database via spoken lyrics. This application is part of a large-scale music database, which is accessed via a graphical user interface complemented with a speech interface. Users can browse a song database indexed by artist album, genre, etc through speech [Gruenstein et al., 2008]. In this work, we extend the song retrieval features to allow query by lyrics. Users who might not recall a song or artist name have the option of querying by speaking lyrics snippets, as in "*and it was all yellow*".

Since the queries are spoken, a front-end ASR decodes the utterance prior to performing song retrieval. The decoded strings of words are then converted into a valid query representation, which is used to search the song database.

We envision such an application deployed on smart hand-held devices, which typically have limited memory and computational resources. Therefore, it is necessary to design speech recognizers with small vocabularies that can be efficiently implemented. Moreover, querying by lyrics is an open-ended task that could involve rare or new words unknown to the ASR. In this research, vocabulary compression and new words are simultaneously addressed by implementing a flat hybrid ASR that decodes a spoken utterance into words and sub-lexical units. In that respect, the research is novel since no previous work has addressed the problem of vocabulary compression for deployment on mobile devices or the usage of a hybrid front-end ASR.

A flat hybrid ASR is constructed by manually omitting words in the recognizer lexicon based on their frequency in the language model (LM) training data. This effectively also manipulates the OOV rate of the recognizer on the LM data. Omitted words are then replaced in the LM data with their sub-word representation. In this research, multiple hybrid ASRs are implemented over a range of vocabulary sizes and associated OOV rates. In order to evaluate the song retrieval model, data collection was performed where users were prompted to record and transcribe lyrics snippets. The recorded data is used to query a 37k-song database.

The questions that are addressed in this chapter are:

1. What are the approaches to querying a text with a spoken utterance?

2. What is the upper-bound on the retrieval performance, when reference text transcriptions are used instead of spoken queries?

3. How does the retrieval performance vary as a function of OOV rate?

4. How do the ASR language model order and the database index size impact the retrieval performance?

5. How does the performance of the sub-word based hybrid ASR model compare to that of other units such as phonemes?

6. How does the performance of the hybrid ASR compare to that of a word-only ASR?

7. What are the effects of implementing database indices that are word only, hybrid (word and sub-word), or sub-word only.

7.2 Related Work

The area of spoken query processing is fairly recent. However, there has been growing interest in the field with the proliferation of increasingly small mobile devices, in-car navigation systems, and automatic directory assistance; all of which have been driving research in this area. For example, speech interfaces have been explored in a movie retrieval system [Moreno-Daniel et al., 2007], an in-car audio data retrieval via metadata [Mann et al., 2007], a cell-phone manual retrieval system [Ishikawa et al., 2004], and a directory assistance model [Natarajan et al., 2002]. A recent article on voice search explored the challenges posed by the technology in areas such as speech recognition and spoken language understanding [Wang et al., 2008].

One of the earliest efforts on document retrieval from spoken query was proposed in [Barnett et al., 2007]. The work reported on the correlation of the retrieval precision with the recognition error rate, the OOV rate, and the query length. The top 1 and top 5 recognition outputs were used to create queries. It was shown that increasing the WER resulted in decreased precision, and that longer queries were more robust to errors than shorter queries. The results could be deemed inconclusive since only 35 queries dictated by a single male speaker were used to evaluate the spoken query document retrieval model. Moreover, the queries were quite long, ranging from 20 to 165 words with an average length of 58 words. The effects of WER on spoken query processing were addressed in more detail in [Crestani, 2000]. This work used the same 35-query set for evaluation purposes. The study showed that the retrieval performance is robust even for high WERs up to ∼40%, particularly for long queries.

Spoken query document retrieval was also addressed in [Wolf and Raj, 2002]. This research tackled the problem of speech misrecognition by incorporating an a posteriori probability weighting scheme for all the words in the lattice generated by the front-end ASR. Moreover, to address the problem of new or OOV words, document keywords were automatically identified in a first pass and were incorporated into the front-end ASR lexicon. The spoken query model proposed in [Wolf and Raj, 2002], was further explored for a business-address finder in [Wolf et al., 2004]. Results indicated that a user interface (UI) complemented with a speech interface was more effective than a menu-based UI. The same model was compared to a menu-based UI in an in-car music retrieval system, and a user-study was conducted on fourteen drivers. Subjects were evaluated on their steering and braking performances as they attempted to search for specific songs. The results indicated that (1) the subjects were better at steering when using the speech interface than when using the menu-based UI; (2) using the speech interface allowed for a faster song lookup; and (3) the brake reaction time was the same for both set-ups.

In [Chang et al., 2002] spoken query information retrieval was implemented on mobile devices for the Chinese language. To account for the linguistic properties of the Chinese language, character and syllable-based indexing were explored. Spoken queries were recorded in three audio channel settings: (1) a headset microphone, (2) a personal digital assistant (PDA) microphone, and (3) a cell-phone microphone. Queries recorded over the cell-phone device yielded the worst results, which was attributed to the lack of matching acoustic training data.

Information retrieval from spoken query was implemented for the Spanish language in [Gonzalez-Ferreras and Cardeoso-Payo, 2007]. A total of 490 queries with a mean length of 16 words were used to evaluate the system. The OOV rate was reduced with a two-pass strategy: (1) the top 1000 relevant documents were retrieved; (2) those documents were used to perform vocabulary and LM adaptation. Foreign words, which were mostly English, were also problematic for the retrieval process. Since the English words were pronounced with a Spanish accent, a mapping from English to Spanish phonemes was developed manually. The mapping was used to provide Spanish-accented pronunciations for the English words in the dictionary.

7.3 The Approach

In this research, the task is to perform song retrieval from spoken lyrics, and, hence, it falls under spoken query processing and requires a front-end speech recognizer. As illustrated in Figure 7-1, the spoken lyrics are first presented to an ASR system which decodes the utterance into one or more sequences of words. The sequences of words are then transformed into a valid query representation which is presented to the information retrieval system. The output of this process is a list of song titles which best match the lyrics query. In the following sections, the components of this process are described in more detail.

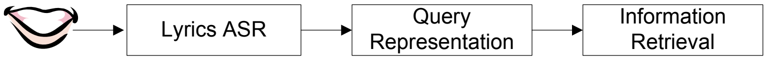


Figure 7-1: A diagram illustrating the information retrieval process. Since the queries are spoken, an automatic speech recognizer is first used to decode the utterances. The ASR output is then transformed into a valid query representation which is used for retrieval.

7.3.1 The ASR System

To build the lyrics ASR system, lyrics for ∼37k songs were collected from *lyricwiki.org* and used as LM training data. The total vocabulary size corresponding to the lyrics LM training data is around 47k. The phonemic dictionary for the vocabulary is looked up in a standard dictionary, and any missing pronunciations are automatically generated using the L2S model as described in Section 3.3. The SUMMIT landmark-based speech recognition system is used in all the experiments [Glass, 2003].

The Sub-Word Based Hybrid ASR

Two goals of this research are to achieve efficient ASR vocabulary compression and to build an open-ended front-end recognizer. These are achieved by designing a flat hybrid ASR similar to the approach in [Bisani and Ney, 2005]. The words in the lyrics vocabulary are listed in ascending order based on their frequency in the LM training data. The OOV rate of the ASR system is then manipulated by keeping only the top *N* most frequent words in the lyrics vocabulary. By varying *N*, different OOV rates can be achieved. The sub-words are inserted into the LM training data by replacing, for each vocabulary size, all the OOV words with their sub-word representation. The sub-words are also added to the ASR lexicon. The resulting OOV model is denoted *flat hybrid* since it predicts and models OOV words simultaneously, guided by a hybrid LM which contains both words and sub-words. The reader is referred to Chapter 2, Section 2.2 for further detail regarding the two most commonly implemented OOV models - flat and hierarchical.

Table 7.1 lists the selected vocabulary sizes and their corresponding OOV rates on the LM training data.

OOV Rate	Vocabulary Size	OOV Rate	Vocabulary Size
50%	68	6%	2766
40%	120	5%	3425
30%	233	4%	4386
20%	492	3%	5890
10%	1443	2%	8572
9%	1666	1%	14532
8%	1942	0%	46937
7%	2294		

Table 7.1: The vocabulary sizes implemented in the recognition and IR experiments and their corresponding OOV rates on the LM training data.

The OOV rate is typically computed as follows:

$$\begin{aligned}
\text{let } A &= \{ \text{ set of all possible unique words } \} \\
\text{let vocabulary } V &\subset A \\
\text{let corpus } C &= (w_1, \ldots, w_{|C|}) \text{ s.t. } w_i \in A \\
\text{let } 1_V(w) &= \begin{cases} 1 & \text{if } w \in V \\ 0 & \text{otherwise} \end{cases} \\
OOV_{rate} &= \frac{\sum_{i=1}^{|C|} 1 - 1_V(w_i)}{|C|} \qquad (7.1)
\end{aligned}$$

Based on the aforementioned description, the output of a hybrid recognizer may contain both words and sub-words as illustrated in Table 7.2 for the three OOV rates, 30%, 10%, and 3%. In these examples, the words *new/beautiful*, *complete*, and *breakable* are not in the 233-, 1443-, and 5890-word vocabularies respectively.

OOV Rate	Sample Recognition Output
30%	i never `n+ -uw+` the world would be so `b+ yu+ tf -ax+ f+ -axl` at all
10%	because of you i felt my life would be `k+ -axm pl+ -iyt`
3%	she had something `br+ -ey+ k+ -ax+ b+ -axl`

Table 7.2: Sample hybrid recognition outputs for three selected OOV rates (30%, 10%, 3%) consisting of strings of words and sub-words.

7.3.2 The IR System: Lucene

The IR toolkit used in these experiments is Lucene, a Java search engine library [Gospodnetic and Hatcher, 2004]. The basic elements of Lucene are the term, field, document, and index.

1. A term is a string, e.g. *Coldplay*.
2. A field is a named sequence of terms, e.g. *Artist: Coldplay*.
3. A document is a sequence of fields, e.g:

    Artist: Coldplay

    Song Title: Yellow

    Lyrics: Look at the stars ... And it was all yellow ...

4. An index is a sequence of documents, and can be naturally viewed as a table which lists, for each document, the terms it contains. However, the index in Lucene is implemented as an inverted index which lists, for each term, the documents that contain it. An inverted index results in a more efficient term-based search. Figure 7-2 illustrates an inverted index as well as its relation to documents, fields, and terms.

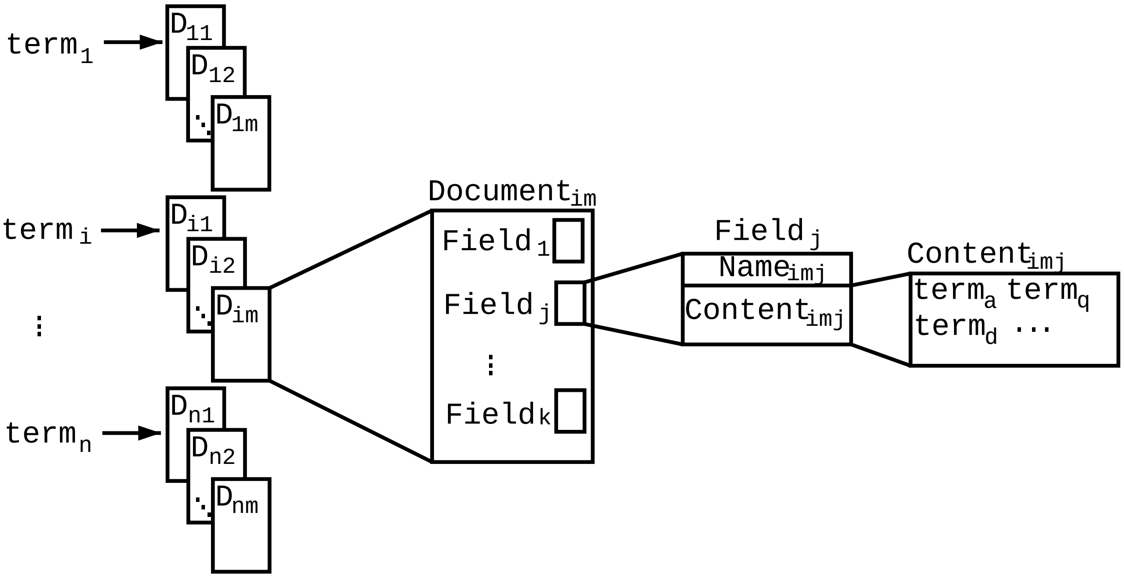


Figure 7-2: Illustration of the inverted indexing implemented in Lucene and the relation of the index to documents, fields, and terms.

In this research, the database is generated with the lyrics of the 37k songs described in Section 7.3.1. Each document is a song, and it consists of fields such as artist, lyrics, genre, album, song title, etc. The lyrics field is tokenized into terms which are then indexed. Terms could be single words or *n*-grams as illustrated in Table 7.3 for the

n-gram order	Terms
1	she had something breakable
2	she had had something something breakable
3	she had something had something breakable
4	she had something breakable

Table 7.3: Term n-grams where n = 1 ... 4 for the lyrics *"she had something breakable"*. Each term is on a separate line.

lyrics *"she had something breakable"*. n-gram terms should be capable of capturing lexical constraints, particularly if the indexing is done in the sub-word space.

Fifteen hybrid indexed databases are generated to match all the vocabulary coverages listed in Table 7.1. For example, for a 90% coverage, 1443 out of ~47k words are preserved in the lyrics data and the rest are replaced with a sub-word representation. The resulting hybrid data are used to generate a song database indexed by hybrid terms that consist of both words and sub-words. A query produced by an ASR with a 90% coverage would be presented to such a database.

Lucene is a search engine that combines Vector Space Modeling (VSM) [Salton et al., 1975] and Boolean Modeling (BM). BM first narrows down the documents that need to be scored based on the Boolean logic in the query representation, and VSM determines how relevant a document is to a query. VSM represents a document as a vector, where each dimension corresponds to a term. If the term occurs in the document, its corresponding dimension in the vector has a non-zero value, which is a function of term frequency and inverse document frequency. Hence, both queries and documents are represented as vectors, and Lucene models the score of a query q for document d as:

$$S(q,d) = coord(q,d)\alpha(q)\sum_{t \in q} tf(t\ in\ d) idf(t)^2 boost(t) \tag{7.2}$$

Where

$$coord(q,d) \propto \frac{|q \cap d|}{|q|} \tag{7.3}$$

$$\alpha(q) = \frac{1}{\sum_{t \in q} boost(t)^2} \tag{7.4}$$

$coord(q,d)$ assigns higher scores to documents that match more terms in query q. $boost(t)$ is the weight of term t and is set to 1 for all the reported results.

The term frequency formula used in Lucene is:

$$tf(t\,in\,d) = (\frac{n_{t,d}}{\sum_k n_{k,d}})^{\frac{1}{2}} \tag{7.5}$$

Where $n_{t,d}$ is the number of times term t occurs in document d, and the denominator is the total number of terms in document d.

The inverse document frequency is also modeled as:

$$idf(t) = 1 + log(\frac{N}{n_t + 1}) \tag{7.6}$$

Where n_t is the number of documents containing term t, and N is the total number of documents.

7.3.3 Query Generation

An ASR system could potentially generate a top-1 hypothesis, an N-best list, or a lattice, which can be compactly represented as a confusion network (CN) [Mangu et al., 2000]. In order to present the output of an ASR system to the search engine, it has to be converted into a valid query representation. A Lucene query is typically a combination of terms and boolean operators, such as `AND` and `OR`. The `AND` operator matches documents that contain all of its operands, whereas the `OR` matches documents that contain either of its operands. In this research, queries are generated from N-best lists. However, for completion we briefly describe CNs.

A CN, informally known as a "sausage", is a directed and weighted graph such that each edge is labeled with a word and its corresponding posterior. Moreover, the sum of the posteriors of all words lying between two nodes of the graph is 1, and the score of any path in the graph is obtained by multiplying the posteriors of all edges in the path. The set of all paths in the original lattice is a proper subset of the set of all paths in the CN.

Table 7.4 and Figure 7.5 illustrate the 10-best and CN outputs of a hybrid recognizer with an OOV rate of 3% for the spoken utterance *"she had something breakable"*. As shown in Figure 7.5, a path from the start to the end node of a CN always includes all the nodes in the graph. Furthermore, all words of a particular sausage are considered to be competing hypotheses.

A sample query composed of 2-gram terms, and generated from the top 1 hypothesis in Table 7.4 is illustrated below:

```
"she had" AND "had something" AND "something br+" AND "-ey+ k+" AND
               "k+ -ax+" AND "-ax+ b+" AND "b+ -axl"
```

The reader is referred to Appendix D for sample queries generated from the 10-best list shown in Table 7.4 and the CN in Figure 7.5.

Following thorough experimentation, it is empirically determined that:

1. The `AND` operator heavily penalizes a document for not containing a term in the query, by completely removing it from the list of possible matches. In this par-

10-best list
she had something `br+ -ey+ k+ -ax+ b+ -axl`
is she had something `br+ -ey+ k+ -ax+ b+ -axl`
as she had something `br+ -ey+ k+ -ax+ b+ -axl`
she had something `br+ -ey+ k+ -ax+ b+ -axl +z`
she had something `br+ -ey+ k+ -ax+ b+ -axl -d`
she had something `br+ -ey+ k+ -ax+ b+ -axl -iy+`
she had something `br+ -ax+ k+ -ax+ b+ -axl`
she had something `br+ -ey+ k+ -er+ b+ -axl`
she had something `br+ -eh+ k+ -ax+ b+ -axl`
verse you had something `br+ -ey+ k+ -ax+ b+ -axl`

Table 7.4: The 10-best output of a hybrid recognizer with a 3% OOV rate for the utterance *"she had something breakable"*.

ticular application, such a penalty is not recommended, since words in a query can be misrecognized by the ASR. In other words, it is unfair not to consider a document as a possible match because it fails to contain a misrecognized query term. For this reason all `AND`s are replaced with `OR`s.

2. If a term, t, occurs M times in the 10-best list ($M < 10$), its corresponding score in Equation 7.2 will be counted M times. Hence, Equation 7.2 inherently boosts terms based on how frequently they occur in the 10-best list.

3. No gain is observed in using N-best lists of sizes larger than 10, for example 20.

4. Queries generated by CNs produce worse results than those generated by 10-best lists. This can be attributed to the confusion introduced by a CN, which models substantially more competing hypotheses than a 10-best list.

5. No gain is observed in using the posteriors obtained by the CNs as term boosters in Equation 7.2.

Therefore, in this research, queries are generated from 1 and 10-best lists as follows: (1) Each recognition output is converted into a sequence of n-gram terms which are combined with `OR`s; (2) in the case of a 10-best list, the queries corresponding to each recognition output are again combined with `OR`s.

7.4 Data Collection

In order to evaluate the song retrieval system, data collection is conducted as follows. 1k songs are selected from the 37k-song database, and divided into groups of 50. Twenty subjects (13 males and 7 females) are instructed to listen to 30-second snippets of 50 songs each, and to record any portion of the lyrics that they heard. Subjects were also prompted to transcribe their recordings. The transcriptions are

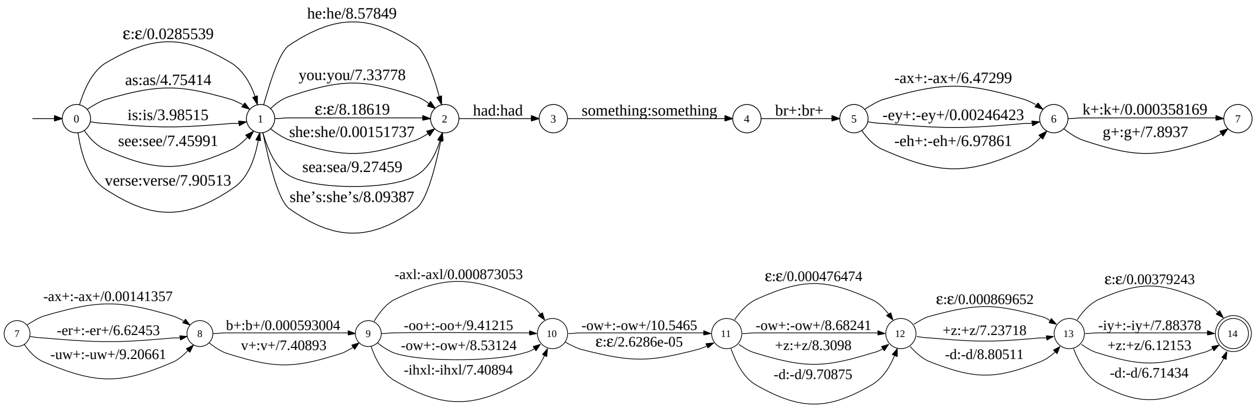


Table 7.5: The confusion network generated by a hybrid recognizer with a 3% OOV rate for the utterance *"she had something breakable"*. The network figure is split in half for lack of space and is read left to right. Note that the confusion network is inclusive of the 10-best list shown in Table 7.4.

treated as reference, and are used to provide an upper-bound on the performance of the song retrieval system. The recordings and the typed text are not error-free since the subjects sometimes misheard words or phrases, replaced contractions, or changed word or phrase order. Moreover, some of the entries are very generic and not beneficial in terms of song retrieval. This is reflected in the results, where, even with the reference queries, a perfect retrieval performance is not obtained. Table 7.6 shows sample problematic queries provided by subjects. For the first three examples, errors are highlighted in *italics* and the correct lyrics are displayed in the second column. The last two rows are examples of uninformative entries that are not likely to yield accurate retrieval.

Typed Lyrics	Correct Lyrics
now i need to step up and be strong	i know i need to step up and be strong
there's no need to cry *and mourn*	there's no need to cry anymore
i'll be *holding* you by new years eve	i'll be over you by new years eve
whiskey	whiskey
la la la	la la la

Table 7.6: Sample problematic queries typed and spoken by subjects during data collection. The first three examples illustrate errors produced by subjects highlighted in *italics* and the corresponding correct version in the right column. The last two examples illustrate generic entries.

Figure 7-3 illustrates the distribution of the length of the submitted queries, where length is defined as the total number of words in the query. The average length of the queries, which ranged from 1 to 48 words, is 8.5.

Next, Figure 7-4 displays, for each vocabulary size in Table 7.1, the OOV rate on

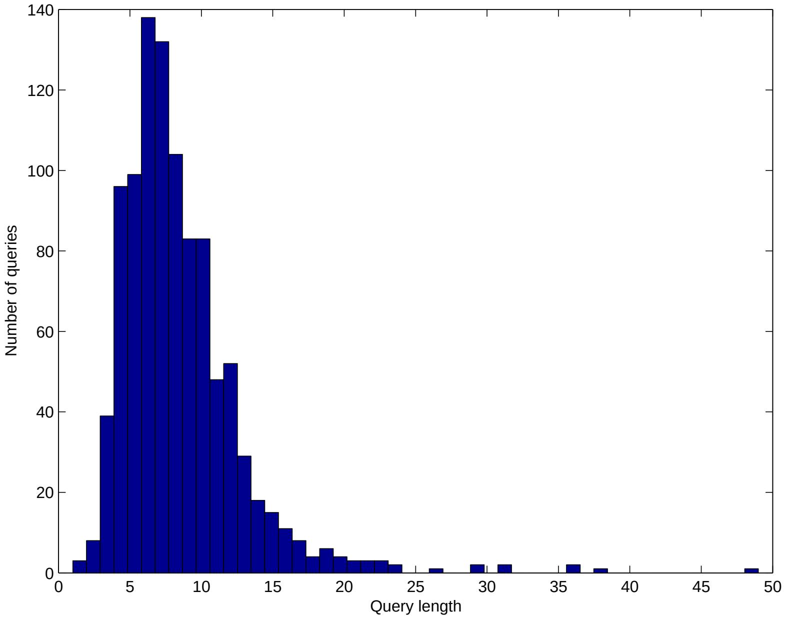


Figure 7-3: The distribution (histogram) of the length of the recorded utterances in terms of number of words.

the LM training data versus that on the collected evaluation data. As illustrated by the plot, the relation between the OOV rates on the two datasets is almost linear. Furthermore, when the OOV rate is 0% on the training data, it is at the non-zero value of 0.2% on the evaluation data. This reinforces the claim that, even with a large vocabulary, a speech recognizer may encounter new words in the test data.

7.5 Recognition Results

As described in Section 7.3.1, the speech recognizer is the first module of the spoken IR process. In this section, we report its performance in terms of Sentence Error Rate (SER), Word Error Rate (WER), and Sub-word Error Rate (SWER). All the results reported in this section are obtained with 3-gram language models (LMs).

7.5.1 Sentence Error Rates (SER)

For every vocabulary size listed in Table 7.1, two speech recognizers are built: (1) a word-only ASR which only contains the words in the vocabulary, and (2) a hybrid ASR which contains all the words in the vocabulary as well as the sub-syllabic sub-word units. Table 7.7 shows sample outputs from both configurations for an OOV rate of 30%, as well as the references. The examples illustrate the role that the sub-word units play when an OOV word is encountered. The hybrid model is still prone to error as shown in the last row; however, the claim is that the sub-word units would still be able to correctly model some of the OOV words, such as "*..-axn my n+ -eym*" shown in the last row of Table 7.7.

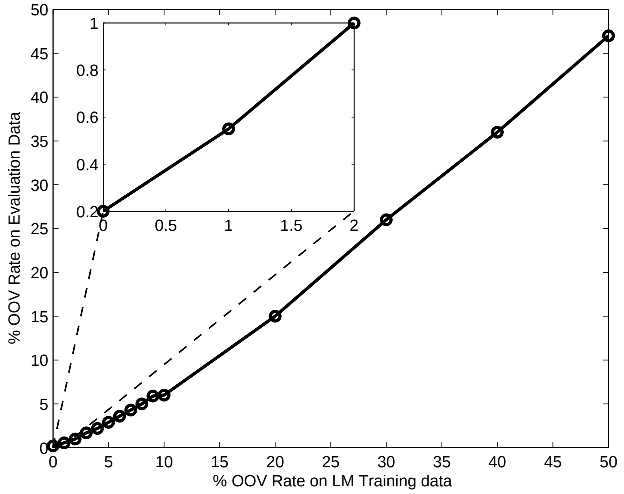


Figure 7-4: OOV rate of the LM training data versus that on the evaluation data over all the implemented vocabulary sizes. The internal plot is a zoom-in on the [0-2%] OOV rate region.

Figure 7-5 compares the SER of the two configurations over all the vocabulary sizes. Both models exhibit an expected decrease in SER as the vocabulary size is increased. Moreover, introducing the sub-word units into the recognizer leads to a decrease in SER compared to the corresponding word-only set-up. However, the difference in SER between the two configurations becomes smaller for larger vocabulary sizes. For example, for a 50% OOV rate, the word-only SER is 99.3%, whereas the hybrid model SER is 88.8%, which is a 10.5% absolute improvement. On the other hand, for a 1% OOV rate, the word-only SER is 67%, and the hybrid SER is 65.3%, which is a 1.7% absolute improvement.

7.5.2 Word Error Rates (WER)

In this section, the WERs of both configurations are analyzed and compared over the range of vocabulary sizes. It is worth noting that utterances containing OOV words might also exhibit errors at the vicinity of the OOV words [Bazzi and Glass, 2000a]. This phenomenon is illustrated in the last example of Table 7.7, "*it for my dreams*". Hence, when analyzing the WER of the hybrid model, it is important to also assess how well the sub-word units are able to correct words in the neighborhood of the OOV words.

As illustrated in Table 7.7 the outputs of the word-only and the hybrid models potentially contain a different number of units. So comparing WERs of the two models would not provide an accurate description of their relative performances. In order to fairly compare WERs, we resort to replacing sub-word sequences in the hybrid model with an <OOV> tag as illustrated in Table 7.8. Such a solution still penalizes the hybrid recognizer for an OOV word even if the prediction is correct and the sub-word

Word-Only ASR	Hybrid ASR	Reference
she had something good girl	she had something `br+ -eyk -ax+ b+ -axl`	she had something *breakable*
i never new the world would be so blue for all	i never `n+ -uw+` the world would be so `b+ yu+ tf -ax+ f+ -axl` at all	never *knew* the world would be so *beautiful* at all
i know to hear it for my dream	i know to hear you `l+ -ihs -axn` my `n+ -eym`	i love to hear you *whisper* my *name*

Table 7.7: Sample outputs from the word-only and the corresponding hybrid ASR as well as the references. The examples illustrate the ability of the hybrid ASR to detect and model OOV words which are highlighted in *italics*.

Word-Only ASR	Hybrid ASR	Reference
she had something good girl	she had something <OOV>	she had something *breakable*
i never new the world would be so blue for all	i never <OOV> the world would be so <OOV> at all	never *knew* the world would be so *beautiful* at all
i know to hear it for my dream	i know to hear you <OOV> my <OOV>	i love to hear you *whisper* my *name*

Table 7.8: Sample outputs from the word-only and the corresponding hybrid ASR where the sub-word sequences are replaced with <OOV>. This replacement is done in order to compare word error rates of the two set-ups.

sequence corresponds to the unknown word. Hence, any improvement in WER is due to the correction of words in the vicinity of the OOV words.

Figure 7-6 illustrates the WER of the word-only and hybrid models, where in the latter, all sub-word sequences are replaced with an <OOV> tag. Similarly to SER, the WER of both configurations decreases consistently as the vocabulary size is increased. Additionally, a significant gain in WER is introduced by the hybrid model when the OOV rate is large ($> 10\%$). This gain decreases to an absolute 1.9% on average for smaller OOV rates ($\leq 10\%$).

7.5.3 Sub-word Error Rates (SWER)

In section 7.5.2, the WER of the hybrid ASR was computed by converting all sub-word sequences to <OOV>. This provided a preliminary comparison of the hybrid ASR to the reference as well as the word-only ASR. To gain a better understanding of the performance of the hybrid ASR model relative the word-only ASR, we report sub-word error rates in this section. The outputs of the word-only and the hybrid ASRs

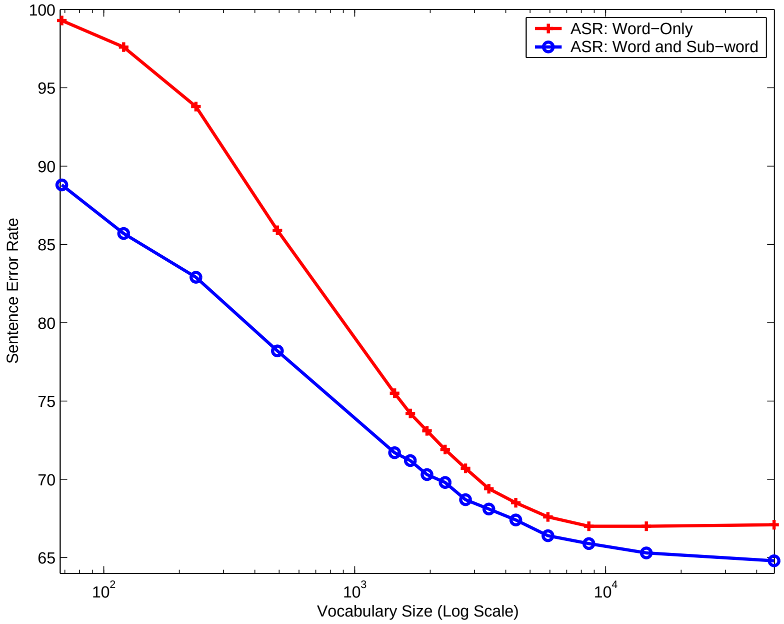


Figure 7-5: The sentence error rates for the word-only and the hybrid ASRs reported over the range of implemented vocabulary sizes.

as well as the reference transcriptions are converted to an all sub-word representation as shown in Table 7.9.

The results are then compared, and sub-word error rates are reported in Figure 7-7.

The results in Figure 7-7 are consistent with those in Figures 7-5 and 7-6 with an improvement in SWER as a function of vocabulary size. Moreover for large OOV rates ($> 10\%$) and corresponding small vocabulary sizes (< 1443 words), a considerable absolute gain in error rate, ranging between 5.3% and 36.3%, is obtained.

7.6 Information Retrieval (IR) Results

7.6.1 Performance Metrics

After reporting speech recognition results in the previous section, IR performance is discussed in this section. IR performance is evaluated in terms of average recall and depth of the correct match. The recall for any particular query is:

$$Recall = \frac{|Relevant \cap Retrieved|}{|Relevant|} \tag{7.7}$$

Relevant is the number of correct (relevant) documents, which could be greater than one, and *Retrieved* is the total number of returned songs, which is fixed to 100 in all the experiments. Hence, recall is the ratio of the total number of relevant songs retrieved by a search over the total number of relevant songs. Note that, if the correct

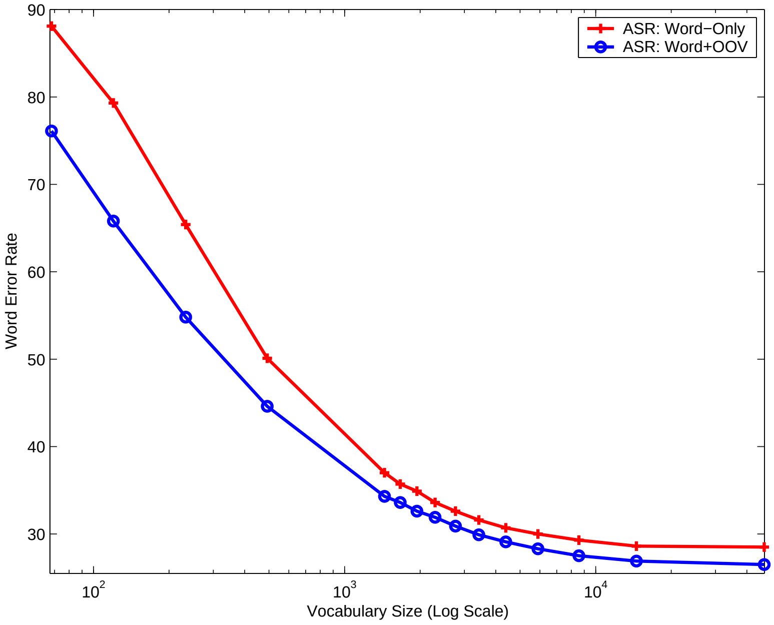


Figure 7-6: The word error rates for the word-only and the hybrid ASRs reported over the range of implemented vocabulary sizes. In the case of hybrid ASRs, sub-word sequences are replaced with the <OOV> symbol prior to computing word error rates.

song is retrieved, $Recall = 1$, otherwise, $Recall = 0$. Figure 7-8 illustrates the space of retrieved and relevant documents as well as their intersection, if any.

7.6.2 Reference Results

As mentioned in Section 7.4, during data collection, subjects were prompted to transcribe the lyrics snippets they recorded. The typed data served as reference, and are used to provide an upper bound on the performance of the IR system. Each of the reference texts is transformed into a query of 1-gram, 2-gram, 3-gram, and 4-gram terms as described in Section 7.3.3. Table 7.10 reports the results in terms of average recall over the 1k songs and for index sizes 1 to 4. As expected from the discussion in Section 7.4, the reference queries do not yield a 100% average recall. The largest improvement in average recall is obtained when increasing the index size from one to two. Increasing the index size to three yields a minor improvement, beyond which we observe a deterioration in average recall.

Figure 7-9 illustrates the cumulative number of matches as a function of depth for all the implemented index sizes. Note that a significant portion of the matches fall below depth 10, e.g., 94% for the 3-gram index. Based on the results in Table 7.10 and Figure 7-9, the 3-gram result is selected as an upper bound for the rest of the experiments.

Word-Only ASR	Hybrid ASR	Reference
sh+ -iy+ hh -aed some th+ -ihng g+ -uhd g+ -erl	sh+ -iy+ hh -aed some th+ -ihng br+ -ey+ k+ -ax+ b+ -axl	sh+ -iy+ hh -aed some th+ -ihng br+ -ey+ k+ -ax+ b+ -axl
i n+ -ehv -er+ n+ -uw+ the w+ -erld w+ -uhd b+ -iy+ s+ -ow+ bl+ -uw+ f+ -aoer+ -aol	i n+ -ehv -er+ n+ -uw+ the w+ -erld w+ -uhd b+ -iy+ s+ -ow+ b+ yu+ tf -ax+ f+ -axl at -aol	n+ -ehv -er+ n+ -uw+ the w+ -erld w+ -uhd b+ -iy+ s+ -ow+ b+ yu+ tf -ax+ f+ -axl at -aol
i n+ -ow+ to hh -ihr it f+ -aoer+ m+ -ay+ dr+ -iym	i n+ -ow+ to hh -ihr you l+ -ihs -axn m+ -ay n+ -eym	i l+ -ahv to hh -ihr you w+ -ihs p+ -er+ m+ -ay+ n+ -eym

Table 7.9: Sample outputs from the word-only and the corresponding hybrid ASR and reference transcriptions where all words are replaced with a sub-word representation. This conversion is done in order to compare sub-word error rates of the set-ups.

Index Size	Average Recall
1	0.852
2	0.887
3	**0.889**
4	0.871

Table 7.10: Average recall for the reference transcriptions as a function of index size.

7.6.3 1-best and 10-best Results

In this section, we compare the IR results for 1-best versus 10-best recognition outputs. All recognition outputs are obtained using a 3-gram LM and are converted into queries of 1-gram and 2-gram terms. In Section 7.6.4, the effect of increasing the index size as well as the LM order, is reported.

Figure 7-10 illustrates the average recall as a function of the implemented vocabulary sizes. Figure 7-10 shows that 10-best outputs generate better results than 1-best, and 2-gram terms perform better than 1-grams. The results also exhibit improvement with the increase in vocabulary size, but it almost plateaus beyond 4.4k. For example, using the whole 47k-word vocabulary produces an average recall of 0.822, while using only a 4.4k-word vocabulary in combination with the sub-word units generates an average recall of 0.806. The relative deterioration in recall is 1.9%, but the relative decrease in vocabulary size is a significant 90.6%.

Figure 7-11 illustrates the cumulative number of correct matches as a function of depth. For clarity only four operating points corresponding to vocabulary sizes of 233, 1.7k, 4.4k, and 47k are shown. The results are obtained with 10-best ASR outputs and 2-gram indices. Similarly to the reference result, which is shown as a bold black solid line, the majority of the correct matches fall below depth 10, e.g.,

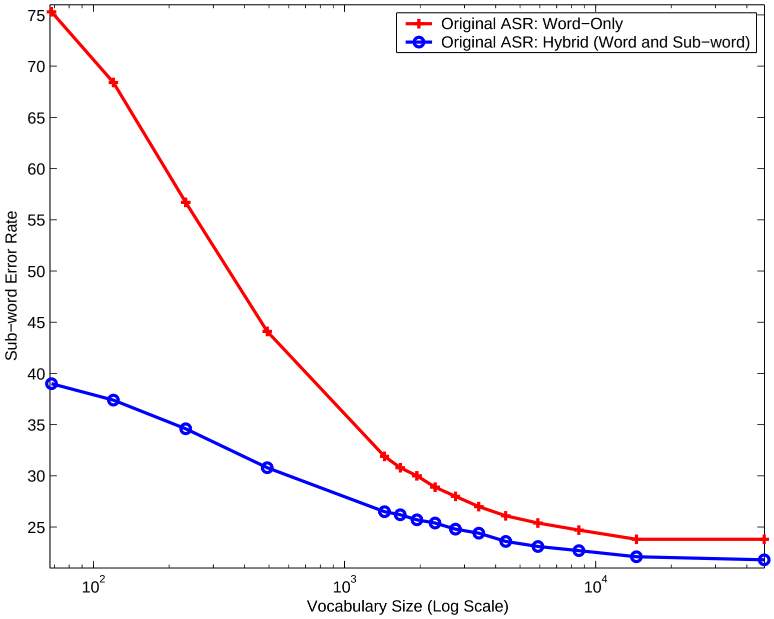


Figure 7-7: The sub-word error rates for the word-only and the hybrid ASRs reported over the range of implemented vocabulary sizes. The sub-word error rate is obtained by converting the ASR outputs and the reference transcriptions into an all sub-word representation and comparing the results.

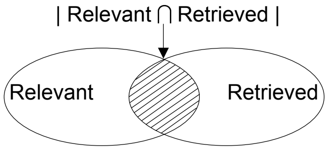


Figure 7-8: An illustration of relevant and retrieved document spaces as well as their intersection, which is shaded. In this research, the number of retrieved documents is always 100.

80% and 88% for the 233-word and 47k-word vocabularies respectively.

7.6.4 Effect of the Index Size and the ASR LM Order

The results of the previous section are obtained with an ASR guided by a 3-gram LM and with 1-gram and 2-gram indices. In this section, the effects of the LM order and the index size are investigated. Figure 7-12 illustrates the average recall as a function of index size and ASR LM order. The evaluated hybrid ASR system has a 492-word vocabulary and a 20% OOV rate. A small-vocabulary hybrid recognizer will often hypothesize sub-word sequences. For this reason, a 492-word ASR is purposefully selected to investigate whether larger index sizes and LM orders, which can model longer sub-word sequences, can yield any performance gain. The results, reported over a range of LM orders, consistently show that most of the gain is obtained when

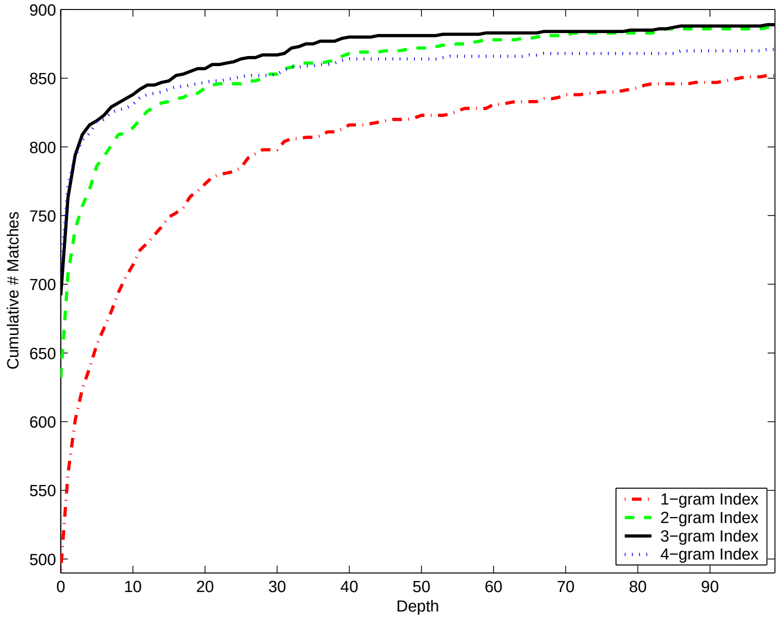


Figure 7-9: The cumulative number of correct matches (out of 1k) as a function of depth (0 to 99) for the reference transcriptions. Results are shown for index sizes 1 to 4.

increasing the index size from one to two. A small gain is obtained for an index of size three, followed by deterioration with size four. As far as the LM order is concerned, we notice that, across all index sizes, the 4-gram LM outperforms the rest. In particular, for a 492-word hybrid ASR, the best average recall is 0.772 and is obtained with a 4-gram LM and an index of size three.

7.6.5 Comparison to Alternative Sub-word Types

So far, we have reported results for the linguistically motivated sub-syllabic units proposed in this thesis. In this section, two hybrid IR systems are implemented with different units, and compared to the sub-syllabic sub-words:

1. phonemes: are the smallest abstract vocal gestures that distinguish words, for example ih, and eh in *bit* and *bet*. There are 61 phonemes in the English language.

2. small sub-words: are based on the original sub-words units except that the units corresponding to the rhyme sub-syllabic structure are further divided into nucleus and coda as illustrated in Figure 3-1, Chapter 3. For example, `-ihng` becomes `-ih! !ng`, and `-eyn` becomes `-ey! !n`. A total of 335 small sub-words are generated from the original linguistically-motivated sub-word units. Table C.1 in Appendix C lists the original rhymes and their decomposition into nucleus and coda.

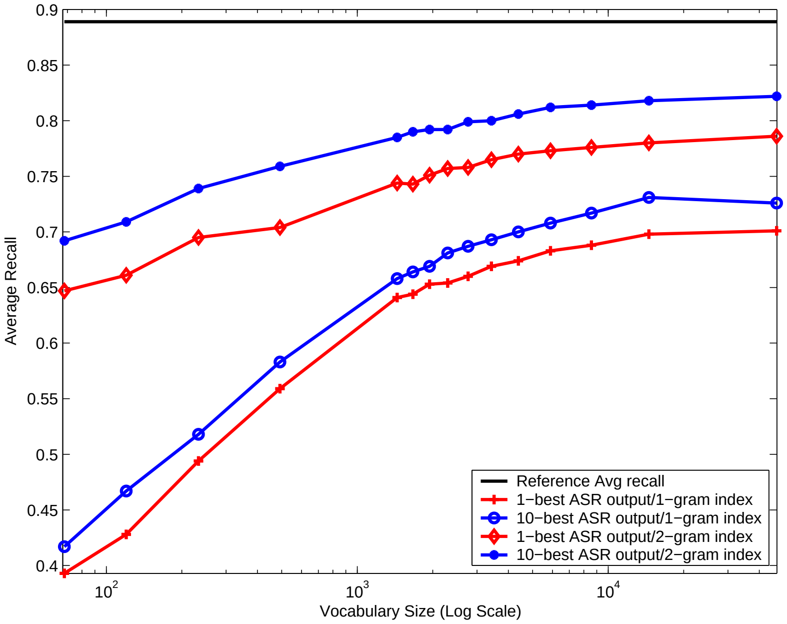


Figure 7-10: The average recall for 1-best and 10-best recognition outputs reported over the range of implemented vocabulary sizes. Results are shown for 1-gram and 2-gram indices. All the ASR systems are built with 3-grams LMs. The best reference result is shown as a black solid line.

Table 7.11 illustrates sample outputs for word-only and hybrid recognizers with a 233-word vocabulary. In this particular example, all three hybrid models, sub-words, small sub-words, and phonemes, generate a perfect representation of the OOV word, *waiting*, whereas the word-only recognizer cannot possibly succeed.

Type of unit	Sample output
word	been *with you* for you oh so long
sub-word	been `w+ -ey+ tf -ihng` for you oh so long
small sub-word	been `w+ -ey+ tf -ih! !ng` for you oh so long
phoneme	been w ey tf ih ng for you oh so long
reference	been *waiting* for you oh so long

Table 7.11: Sample recognition outputs for each of the implemented units: words, sub-words, small sub-words, and phonemes. The outputs are generated with a 233-word recognizer.

Prior to evaluating the different sub-word types, we first check whether the smaller units, such as the phonemes, benefit more from a larger LM order than the original sub-words. This is achieved by comparing the average recall associated with sub-word and phoneme-based hybrid ASRs over LM orders ranging from 2 to 7. Average recall is computed for 2-gram and 3-gram indices, and are reported for a hybrid ASR with a 30% OOV rate (233-word vocabulary) in Figure 7-13. A small-vocabulary ASR is

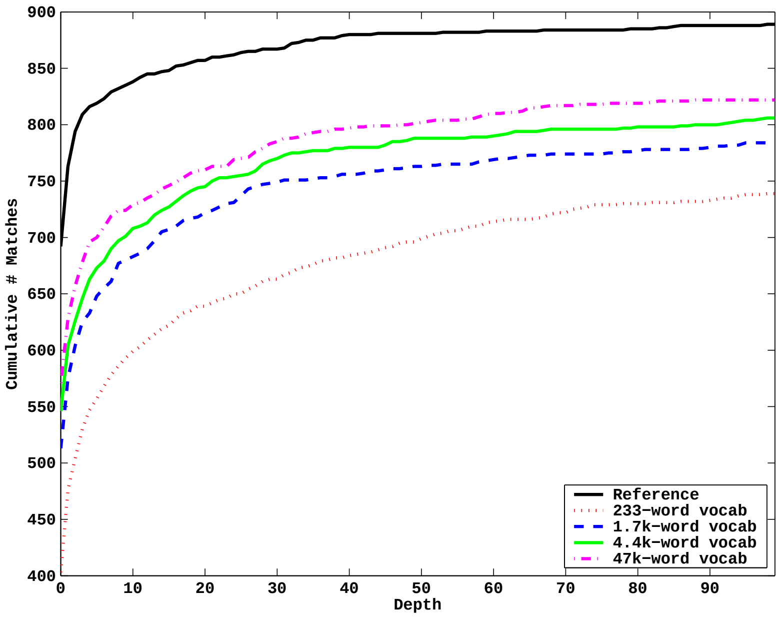


Figure 7-11: The cumulative number of correct matches as a function of depth (0 to 99) for four operating points corresponding to 233, 1.7k, 4.4k, and 47k-word vocabularies. Results are obtained with 10-best ASR outputs and 2-gram indices. The reference result is also shown as a bold black solid line.

purposefully selected for evaluation for the same reason as that cited in Section 7.6.4. It is first noted that the phoneme units do not benefit much from an LM order larger than four. Moreover, the linguistically motivated sub-word units perform better than the phonemes. For example, for a 4-gram LM and 3-gram index, the sub-word based system has an average recall of 0.749, whereas the phoneme-based system has an average recall of 0.707.

Following the comparison of the sub-word and phoneme-based hybrid ASRs as a function of LM order, the average recall associated with the three types of hybrid recognizers: sub-words, small sub-words, and phonemes is evaluated over the range of implemented vocabularies in Figure 7-14. The cumulative number of correct matches is also shown in Figure 7-15 as a function of depth. Again, for clarity, four operating points corresponding to the vocabulary sizes 233, 492, 1.7k, and 4.4k are illustrated. All the recognizers are guided by 4-gram LMs, and 3-gram indices are used. The plots demonstrate that the original sub-words consistently outperform the other units. This could be explained by the fact that the original sub-words are larger than the other units and are, hence, more linguistically constrained. This also explains why the smaller sub-words perform better than the phonemes. Moreover, the difference in performance between the different set-ups becomes smaller with the increase in vocabulary size.

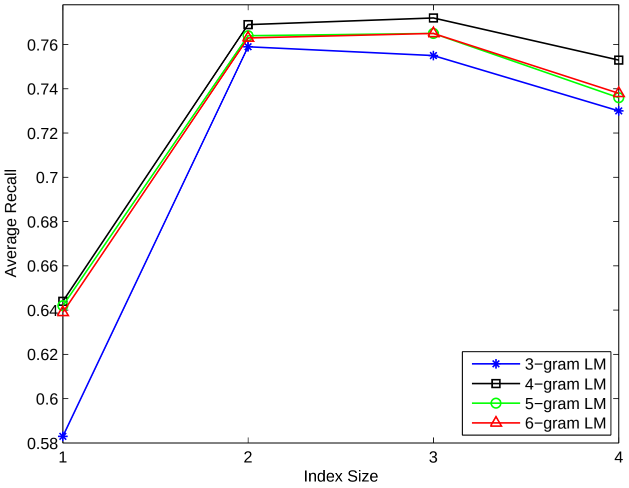


Figure 7-12: The average recall as a function of index size (1 to 4) and ASR LM order (3 to 6). The queries are generated from 10-best recognition outputs. Results are obtained for 492-word hybrid ASRs with a 20% OOV rate.

7.6.6 Comparison to the Word-Only Set-up

In this section, the retrieval performance of the hybrid model is compared to that of a word-only recognition system. We also experiment with a hybrid recognizer in which the sub-word model only serves as an OOV detector. For clarification, we briefly review the first two set-ups and provide a definition for the third:

1. Word-only: the recognizer lexicon and LM only contain words. Hence, the recognition output, and consequently the query, can only consist of words.

2. Hybrid: the recognizer lexicon and LM contain both words and sub-words. The recognition output as well as the query can consist of both words and sub-words.

3. OOV detector: the recognizer lexicon and LM contain both words and sub-words. The recognition output initially consists of both words and sub-words. However, the hypothesized sub-word sequences are used as OOV detectors and are ignored after generating the query. For example, if the query generated by a hybrid ASR is as follows `(word1 word2) (sub-word1 word3) (word4 sub-word2)`, then the corresponding query for the OOV detector model would be `(word1 word2) (word3) (word4)`.

Table 7.12 illustrates sample outputs from the aforementioned set-ups and their corresponding queries consisting of 2-gram terms. Figure 7-16 compares the hybrid, word-only, and OOV detection set-ups in terms of average recall as a function of implemented vocabularies. Figure 7-17 illustrates the cumulative number of correct matches as a function of depth for the four operating points 233, 492, 1.7k, and 4.4k-word vocabularies. The recognizers all are guided by 3-gram LMs and 2-gram

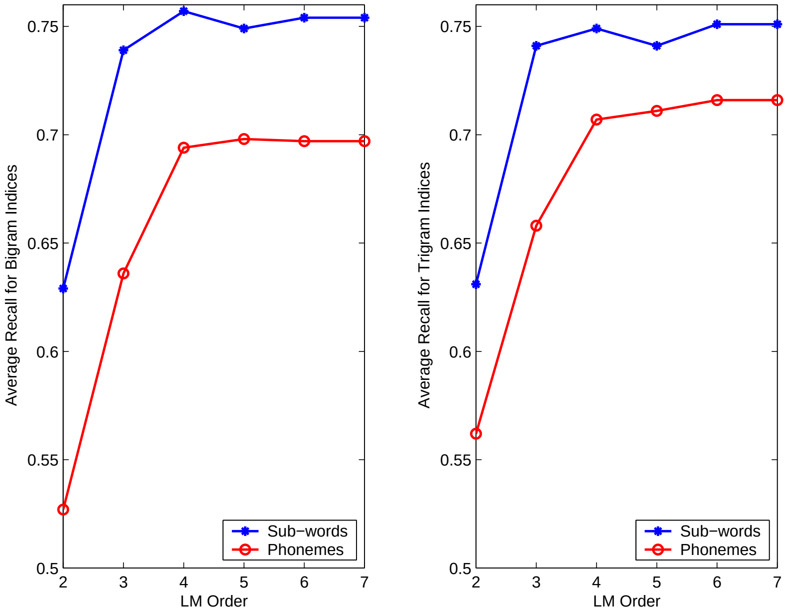


Figure 7-13: Comparison of the original sub-words and phonemes in terms of average recall as a function of LM order. The results are reported for 2-gram (left) and 3-gram (right) indices. The queries are generated from 10-best recognition outputs. The results shown are for 233-word ASR systems with a 30% OOV rate.

indices are used. For small vocabularies, the IR process significantly benefits from the sub-word model. For example, for a 233-word vocabulary with a 30% OOV rate, the average recalls of the hybrid, OOV detection, and word-only models are 0.739, 0.542, and 0.434 respectively. Furthermore, using the sub-word model for OOV detection only is advantageous over not using it at all, as in the word-only model. This is possibly due to the fact that the OOV detection model reduces the errors in the vicinity of the OOV words as mentioned in Section 7.5.2.

7.6.7 Sub-word Based Indexing

In Sections 7.6.2 through 7.6.6, database indexing was either word-based or hybrid - including both words and sub-words. In this section, we investigate the effect of implementing a sub-word only database index, and we compare its retrieval performance to that of a hybrid index. To generate the sub-word based index, the lyrics are first converted to an all sub-word representation. Then, for every sub-word based term, the list of songs that contain it is generated. In this section, we report results for 4-gram ASR LMs and 3-gram database indices. To obtain a valid sub-word based query, the ASR outputs are converted to an all sub-word representation as described in Section 7.5.3. Figure 7-18 compares the average recall of the sub-word only index to the hybrid index over the implemented vocabulary sizes. It is important to note that the sub-word vocabulary used in this section underwent minor modifications compared to the previously described experiments in this chapter. Hence the results

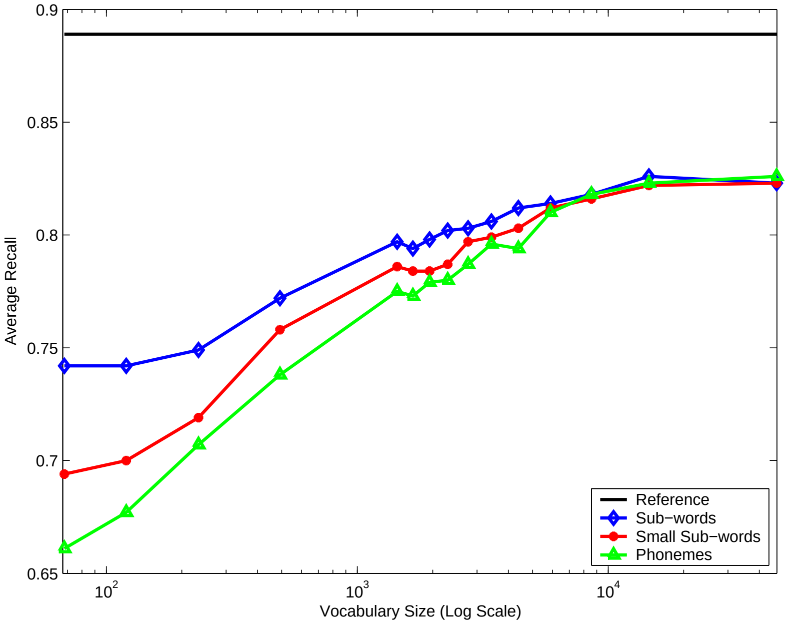


Figure 7-14: The average recall for the sub-word, small sub-word, and phoneme based hybrid ASRs. The results are reported over the range of implemented vocabulary sizes, and are obtained with 4-gram LMs and 3-gram indices.

for the hybrid database index are slightly different from those reported in Figure 7-16. The results in Figure 7-18 indicate that, for all except the 68-word front-end ASR, the sub-word only index performs similarly or improves upon the hybrid index. The improvements can be attributed to the pure sub-word representation, which rewards partially correct words. A significant advantage to using a sub-word only database index as opposed to a hybrid one is the ability to accommodate any front-end recognizer irrespective of the OOV rate. For example, whether the front-end recognizer has an 1%, 10%, or 50% OOV rate, the recognition output can be converted to an all sub-word representation and presented to the same sub-word only indexed database.

7.7 Summary and Discussion

In this chapter, we addressed the problem of song retrieval from spoken lyrics. A continuous ASR is implemented as a front-end to an indexed database. Vocabulary compression and open-ended query recognition are achieved by designing a flat hybrid ASR capable of hypothesizing strings of words and sub-words. To account for uncertainty in the recognition output, *10*-best lists are examined as well as *1*-best outputs. The recognition outputs are converted into a valid query representation prior to searching the song database.

The performance of the front-end recognition system is reported in terms of sentence, word, and sub-word error rates. The hybrid ASR is shown to outperform a

Set-up	Recognition Output	2-gram Query
Word-Only	she had something good girl	(she had) (had something) (something good) (good girl)
Hybrid	she had something `br+ -eyk -ax+ b+ -axl`	(she had) (had something) (something `br+`) (`br+ -eyk`) (`-eyk -ax+`) (`-ax+ b`) (`b -axl`)
OOV Detector	she had something `br+ -eyk -ax+ b -axl`	(she had) (had something) (something)

Table 7.12: The queries composed of 2-gram terms and generated for each of the three recognition set-ups, word-only, hybrid, and OOV detector for the utterance "*she had something breakable*".

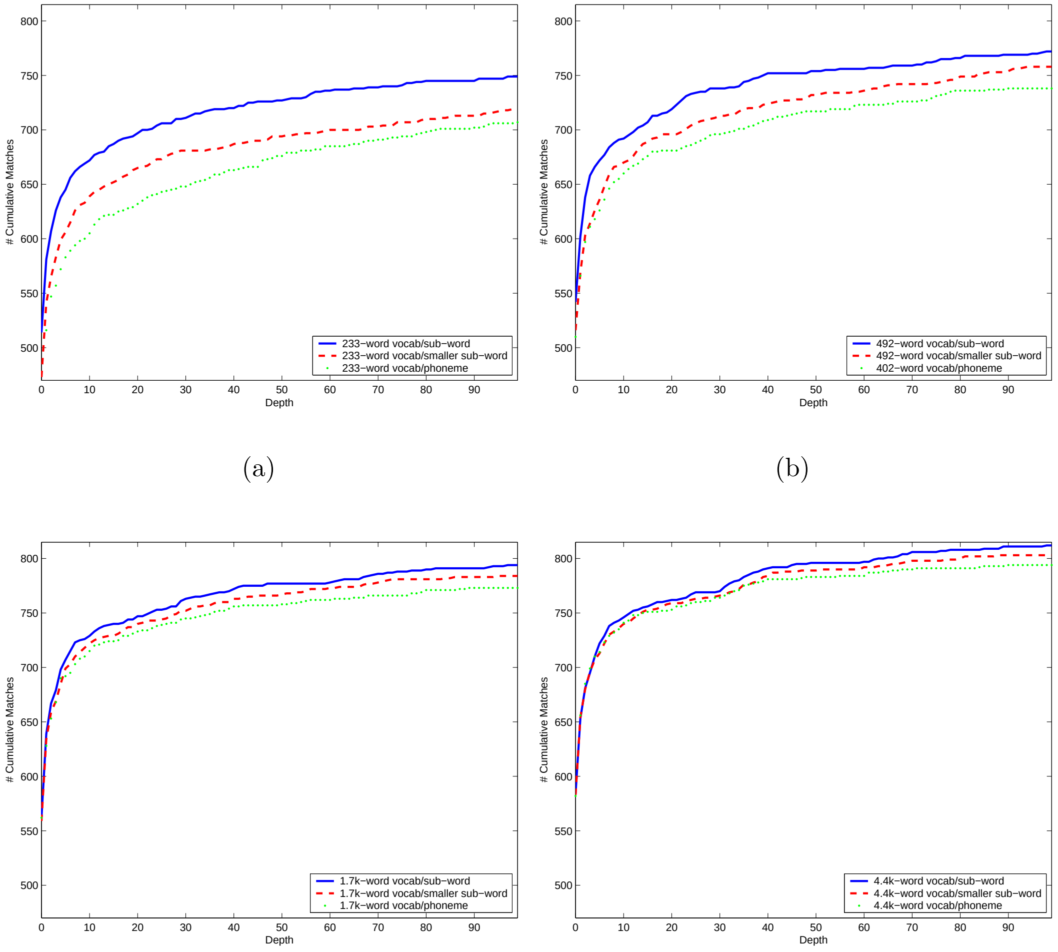
word-only system over a range of out-of-vocabulary rates (1%-50%) with the gain being significant for large OOV rates (>10%).

The retrieval performance is also assessed as a function of ASR *N*-best size, language model order, and the index size, which are set to 10, 4, and 3 respectively following a thorough empirical study. Moreover, the sub-words outperformed alternative linguistically-motivated sub-lexical units such as phonemes. In the future, we aim to compare the sub-words to graphones, which are hybrid units generated using a data-driven approach [Bisani and Ney, 2002, 2008].

We observed that when the vocabulary size is dramatically compressed, the hybrid model suffers little loss in performance. For example, as shown in Figure 7-19, a reasonable operating point is with a 4.4k-word hybrid ASR guided by a 4-gram LM. As illustrated in Figure 7-19(a), at that operating point, a minor 1.3% loss in average recall is observed compared to the 47k-word ASR. Moreover, Figure 7-19(b) shows that, at a depth of 10, 743/1000 songs are correctly retrieved with the 4.4k-word model versus 766/1000 for the 47k-word ASR. This corresponds to a small 3% deterioration in performance. On the other hand, the vocabulary has been compressed by more than a factor of 10. A small vocabulary could be an important consideration for implementation on a hand-held device.

We also implemented and compared three types of database indices: (1) a word only; (2) a hybrid; and (3) a sub-word only. We observed that a sub-word only index had the best performance, possibly since the sub-lexical representation rewards partially correct terms.

Although the spoken retrieval model presented in this chapter was implemented in the music domain, one can envision various scenarios where the model can be advantageous such as a news article browser or a directory assistance application. Moreover, a speech modality for a retrieval system is appealing in many situations. For example, it is a convenient medium to access hands-free speech-enabled systems used in vehicles. Speech interfaces can also enhance keyboard interaction with hand-held devices, which are becoming increasingly small.



(a)

(b)

(c)

(d)

Figure 7-15: The cumulative number of correct matches as a function of depth (0 to 99) for the four operating points 233, 492, 1.7k, and 4.4k-word vocabularies, which correspond to Figures 7-15(a), 7-15(b), 7-15(c), and 7-15(d) respectively. The results are obtained with 4-gram LMs and 3-gram indices, and are plotted for phonemes, small sub-words, and sub-words.

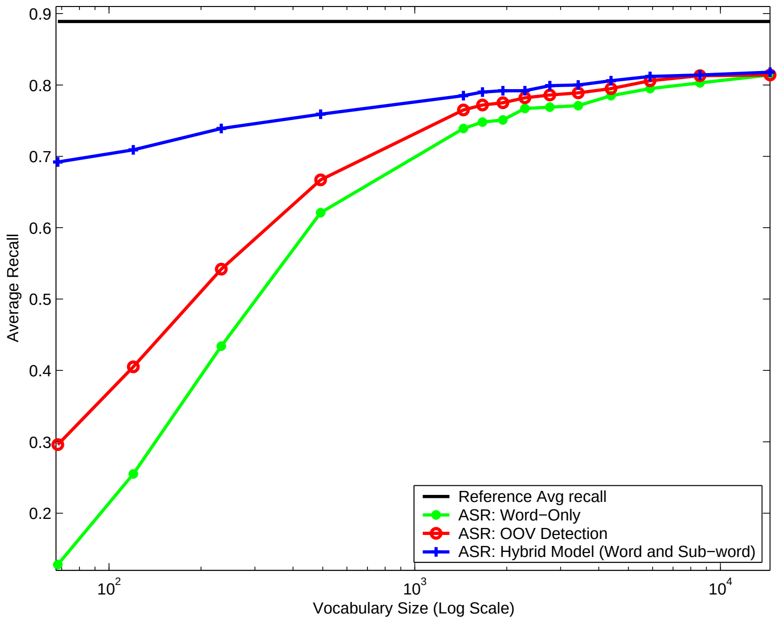
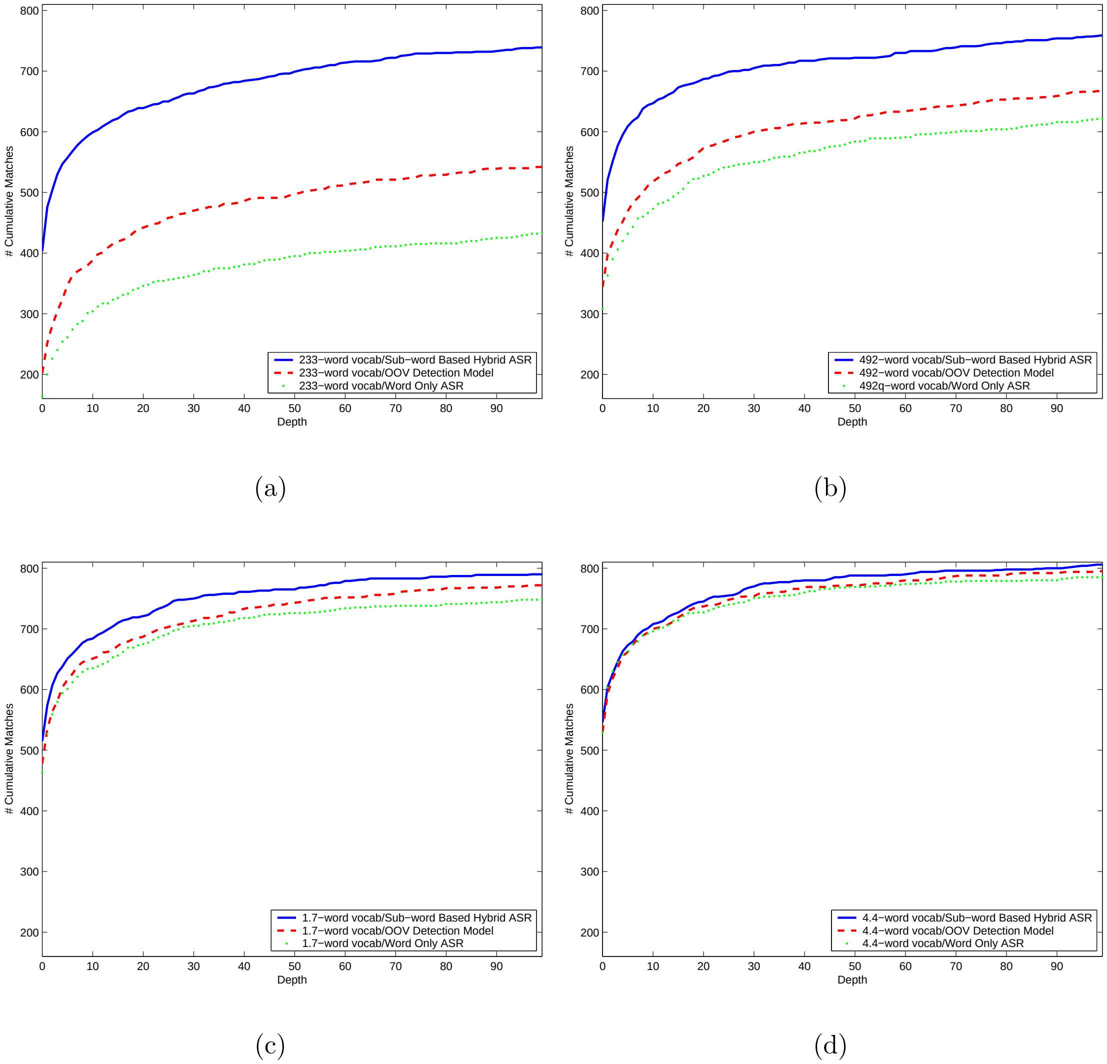


Figure 7-16: The average recall for three ASR models: (1) a word-only; (2) an OOV detection; and (3) a hybrid model. The OOV detection model operates by using a hybrid ASR front-end, and ignoring any hypothesized sub-word sequences during retrieval. The results are reported over the range of implemented vocabularies, and are obtained using 3-gram ASR LMs and 2-gram indices.



(a) (b)

(c) (d)

Figure 7-17: The cumulative number of correct matches as a function of depth (0 to 99) for the four operating points 233, 492, 1.7k, and 4.4k-word vocabularies, which correspond to Figures 7-17(a), 7-17(b), 7-17(c), and 7-17(d) respectively. The results are obtained with 3-gram LMs and 2-gram indices, and are plotted for the sub-word based hybrid ASR, the OOV detection model, and the word only ASR.

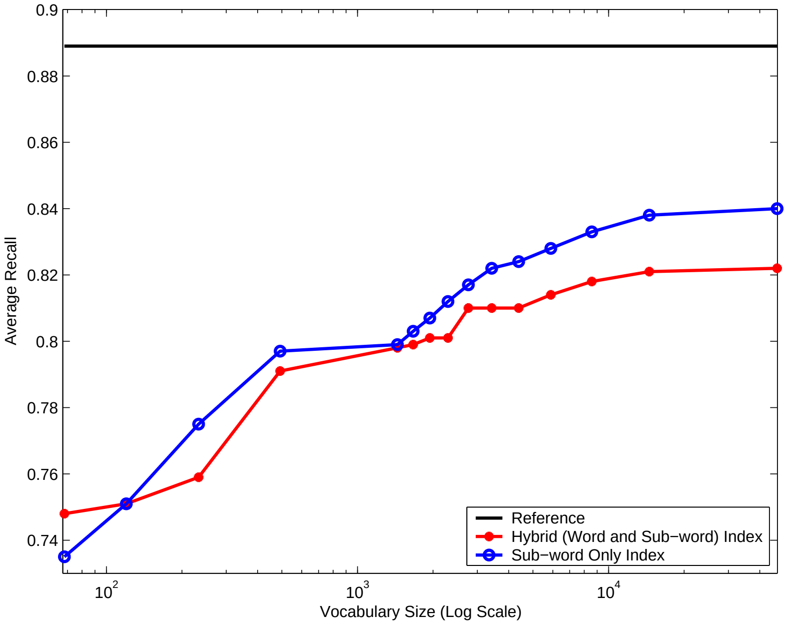
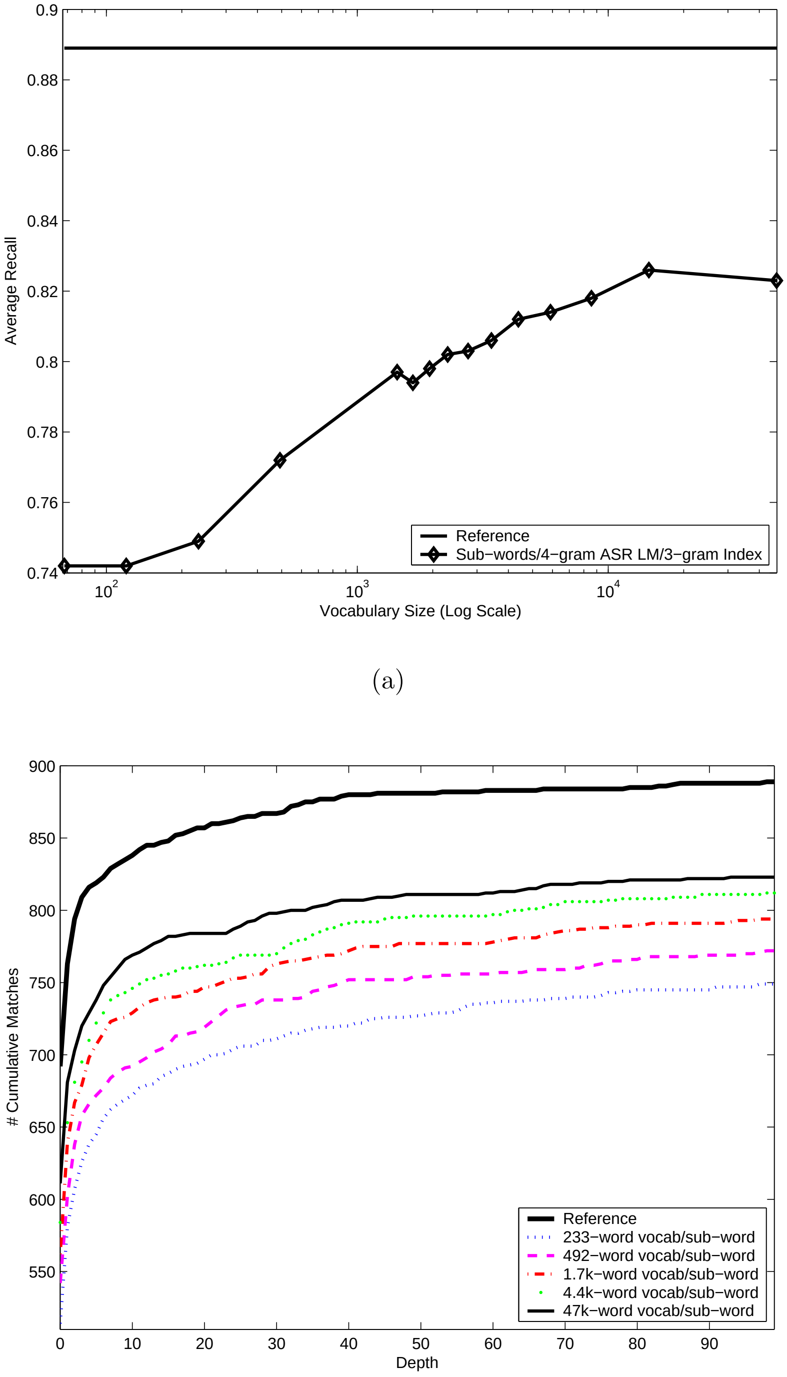


Figure 7-18: A comparison of the retrieval performance for the hybrid versus sub-word only database index. Average recall is reported as a function of the implemented vocabulary sizes. The sub-word vocabulary used to generate this plot underwent minor modifications compared to the previously described experiments in this chapter. Hence the results for the hybrid database index are slightly different from those reported in Figure 7-16.



(a)

(b)

Figure 7-19: Figure 7-19(a) illustrates the best recall results obtained with the hybrid model as a function of the implemented vocabulary sizes using a 4-gram LM and 3-gram indices generated from 10-best recognition outputs. Figure 7-19(b) is a plot of the cumulative number of correct matches as a function of depth (0 to 99) for the five operating points 233, 492, 1.7k, 4.4k, and 47k-word vocabularies.

Chapter 8

Summary and Future Work

8.1 Summary

In this thesis, we addressed the problem of sub-word modeling, which involves representing words with sub-lexical units. We argued that such a model could be advantageous in a number of speech recognition applications. For example, a sub-word recognizer could learn the pronunciation or spelling of a word, depending on whether the sub-word units encoded pronunciation or graphemic knowledge or both. Moreover, if a word-based ASR is augmented with a sub-word model, either within a serial or parallel configuration, it would be able to handle new or out-of-vocabulary words. In this section, we summarize the contributions and outcomes of this thesis.

An Overview of Previous Research

Addressing The New Word Problem

One of the motivations for sub-word modeling is the ability to model *any* word, including new words, with strings of sub-lexical units. Hence, sub-word modeling is a potential solution to the new or out-of-vocabulary (OOV) word problem. Through a thorough literature review, we demonstrated that the new word problem is an inevitable challenge that faces ASR. We claimed that, in order to handle OOV words, ASR should undergo a paradigm shift from vocabulary design to that of using more intelligent models that can detect and learn new words. We then presented a detailed overview of previous work on OOV modeling, which includes (1) OOV word detection, and (2) OOV word learning. We reviewed the two most common approaches towards OOV modeling: (1) the filler model, that can be viewed as a hierarchical or parallel approach towards OOV modeling, where first the OOV word is detected, and then it is modeled using some form of sub-lexical representation; and (2) the flat hybrid model, which can be viewed as a serial approach that detects and models OOV words simultaneously. Finally, we showed that learning an OOV word involves the update of the ASR dictionary with a lexical entry, and this is tied with sub-word modeling.

Letter-to-Sound/Sound-to-Letter Modeling

Letter-to-sound/sound-to-letter (L2S/S2L) modeling is concerned with the process of converting symbols from one domain to another, namely pronunciation to spelling and vice versa. Such a process involves learning an alignment between sound and graphemic units and this inherently goes hand in hand with sub-word modeling. The L2S/S2L models proposed in previous research spanned data-driven and linguistic approaches as well as sub-lexical units that modeled: (1) phonemes, (2) phoneme clusters, (3) letters, (4) or combinations of phonemes and letters. The developed S2L/L2S models were successfully evaluated on grapheme-to-phoneme and phoneme-to-grapheme conversion for English dictionaries [Galescu and Allen, 2001; Bisani and Ney, 2002], proper names [Galescu and Allen, 2002], and foreign dictionaries such as German and French [Bisani and Ney, 2002, 2008].

The Sub-Syllable as a Sub-Lexical Unit

The sub-word units proposed are sub-syllabic in nature. Our choice was motivated by several phonological theories and seminal doctoral theses that argued that the syllable is a critical linguistic unit that can account for a number of crucial phonological aspects such as phonotactic constraints, stress, and tone as well as certain phonological phenomena, such as /t/-flapping and /r/-insertion and deletion. Syllabic and sub-syllabic units have also been slowly emerging as basic recognition units in ASR instead of phonemes. The claim is that syllables and sub-syllables are more reliable than phonemes since they are larger linguistically-motivated units capable of capturing phonotactic constraints and higher-level prosodic knowledge.

Linguistically-Motivated Sub-word Modeling

We proposed sub-words based ASR instead of the conventionally used word-based model. The sub-word units presented in this research encode only pronunciation information and can be considered agglomerations of one or more phonemes. They were primarily designed using context-free rules that encode sub-syllabic linguistic knowledge such as positional and phonological information. The grammar also made use of sonority rules within a syllable combined with the maximal onset principle to make informed decisions about syllable boundary locations. The grammar consisted of four hierarchical layers: (1) The root node, which consisted of a word; (2) the second layer defined the sub-syllabic structure of English words; (3) the third layer defined all possible ways sub-syllables can be pronounced in terms of sub-word units; and (4) the fourth layer described all possible ways sub-words can be spelled. The grammar, which was derived from training data through a boot-strapping procedure, was supported by a probability model, which enhanced the context-free rules with scores based on frequency of usage in a large training set. The grammar parsed words using a best-first search strategy guided by the probability model. Though the proposed sub-word model initially required some manual labor, its appeal lies in its relative simplicity (a four-layer grammar supported by a probability model), and the elegance of the notation scheme, which tied directly to a phoneme notation typically used in phoneme-based speech recognizers.

A Letter-to-Sound Model

The alignment between the third and fourth layers in the proposed grammar gave rise to hybrid units denoted as spellnemes, which encoded both spelling and pronunciation knowledge. We leveraged these hybrid units to build a bi-directional L2S model. We described in detail the design of the spellneme units, as well as the implementation of the L2S model using finite state transducers (FSTs). At the core of the L2S model was a spellneme language model which was trained on spellneme sequences obtained by parsing a set of words through the grammar. The L2S model was extensively evaluated on the task of automatic lexical learning. The S2L model was implemented in a lexical access algorithm, as a back-end to a sub-word recognizer, which converts novel phonemic sequences to a valid graphemic representation.

Automatic Lexical Learning

We presented a thorough empirical study on automatic lexical learning using the L2S and S2L models. In the first set of experiments, perfect knowledge of the spelling was assumed, and two approaches were proposed for automatically generating lexical entries: (1) the L2S model, which converted letter sequences into a valid phonemic representation; and (2) a sub-word recognizer, which decoded spoken words into sub-word strings that were converted to phonemic pronunciations. The generated lexical entries were evaluated on an isolated word recognition task, and the following results were noted: (1) the lexical dictionary automatically generated with the L2S model was comparable in performance to a dictionary that was manually edited by lexical experts; (2) initial improvement in recognition performance was observed as more alternative phonemic pronunciations were incorporated by the L2S model into the dictionary. However, the performance eventually degraded as pronunciation confusion was increased, in this case beyond 20 pronunciations; (3) when pronunciations generated from spoken data and a sub-word recognizer were combined with L2S-based pronunciations, further improvement in recognition performance was observed.

In the second set of experiments, the assumption of perfect spelling knowledge was relaxed and a lexical entry was learned from spoken renderings of a word and its spelling. We proposed an iterative and unsupervised algorithm, which presents spoken instances of both spellings and words to a letter and sub-word recognizer respectively. The output of each recognizer was then processed by a bi-directional L2S model and injected back into the other recognizer in the form of *soft* language model bias. The algorithm was denoted as *Turbo-style* in reference to Turbo Codes, which follow the same feed-back loop principle. The algorithm was evaluated in terms of spelling accuracy, letter error rate, and phonetic error rate of the lexical entries. The automatically generated lexical dictionaries were also evaluated on an isolated word recognition task in terms of word error rate. Following evaluation, the following was observed: (1) the spelling accuracy and the letter error rate of the generated lexical entries exhibited significant absolute improvements following only two Turbo iterations; (2) the pronunciation accuracy and the phonetic error rate of the learned pronunciations also showed similar trends; (3) the phonemic dictionary obtained following two iterations of the Turbo algorithm significantly outperformed a manually transcribed dictionary on an isolated word recognition task.

A Parallel Hybrid ASR Model

We evaluated the sub-word units in an isolated word recognition task by embedding a sub-word recognizer in a parallel fashion as a back-off model for a word recognizer. The resulting hybrid model was evaluated in a lexical access application where a user spoke a word and the word recognizer hypothesized and displayed the top candidate words. If the correct word was not in the returned list, the system triggered the sub-word recognizer. In the process of building the hybrid model, several aspects of the sub-word model were assessed: (1) the performance of the sub-word recognizer in the error recovery system was evaluated in isolation as a function of sub-word language models and *N*-best list depth; (2) in order to properly evaluate the open-ended hybrid model, it was compared to a large-vocabulary recognizer. In the process of building a large-vocabulary recognizer, a phonemic dictionary corresponding to the lexicon was automatically generated using the L2S model. Hence, in this research, we further evaluated the ability of the L2S model to automatically generate phonemic pronunciations. The parallel hybrid model was able to correctly recover OOV words in its top-10 output 69% of the time. Moreover, it outperformed the large-vocabulary recognizer on an isolated word recognition task.

A Flat (Serial) Hybrid ASR Model

In this research, we also evaluated the sub-word units in a continuous flat hybrid ASR. The model was denoted as “flat hybrid” since it predicted and modeled OOV words simultaneously guided by a hybrid LM which contained both words and sub-words. The flat hybrid ASR was designed as follows: (1) a set of words was purposefully omitted from the ASR lexicon, hence manipulating its OOV rate on the language model training data; (2) the sub-lexical units were integrated into the LM training data by replacing all the OOV words with their sub-word representation; (3) the sub-lexical units were added to the ASR lexicon. The hybrid ASR was implemented as an open-ended lyrics recognizer, which was used as a front-end to a song retrieval system. To account for uncertainty in the recognition output, *N-best* lists were examined as well as *1*-best outputs. The song retrieval model was evaluated in terms of (1) speech recognition performance of the front-end ASR; and (2) retrieval performance of the overall system. The performance of the front-end recognition system was reported in terms of sentence, word, and sub-word error rates. The hybrid ASR was shown to outperform a word-only system over a range of OOV rates. The retrieval performance was assessed as a function of ASR *N*-best size, language model order, and the index size. Following an empirical study, a 10-best recognition output was generated guided by a 4-gram language model. The recognition output was postprocessed to generate a valid query representation and was presented to a database of index size 3. Moreover, the linguistically-motivated sub-word units outperformed other types of sub-lexical units such as phonemes. We observed that even with a dramatic reduction in vocabulary size (by more than a factor of 10), the hybrid model suffered only minor loss in retrieval performance. Vocabulary compression becomes of interest if the retrieval model were to be deployed on small footprint devices.

8.2 Future Work

We have presented a linguistically-motivated sub-word model and extensively evaluated it over a range of applications. Different directions can be taken to extend and improve this work. In this section, we propose various extensions to the research developed in this thesis.

Sub-word Acoustic Modeling

In this research, the sub-word units were incorporated into the LM of an ASR. Previous work has demonstrated gain in performance from integrating sub-lexical units into a speech recognizer from an acoustic modeling perspective [Hausenstein, 1997; Wu et al., 1998a,b]. Speech utterances were automatically segmented into sub-lexical units larger than phonemes such as syllables, and acoustic models were trained on measurements extracted over the segments. In [Wu et al., 1998a], it was argued that speech intelligibility shows dependence on relatively slow changes of 2-16 Hz in the spectrum of the speech signal, and the suppression of modulations in the 28 Hz range significantly degraded speech intelligibility. This claim motivated the use of syllables as the basic recognition units instead of phonemes, since modulations in this frequency range (2-16Hz) are associated with the typical durations of syllables.

Data-Driven Approaches Towards Sub-word Modeling

The design of the sub-word units was guided by phonological and linguistic knowledge. Previous research have explored the automatic generation of sub-lexical units using data-driven approaches [Deligne and Bimbot, 1997; Bisani and Ney, 2002, 2008; Galescu and Allen, 2001]. A possible extension to this research is a merge of the two approaches where syllabic and sub-syllabic structure can be automatically discovered. Since, in this research, the sub-word units were developed within the context of a context-free grammar, one possible approach towards automating this process is through grammar induction [Duda et al., 2000]. The grammar describing sub-syllabic structure could be inferred from a set of observations.

On another level, it would also be valuable to conduct a thorough empirical study comparing linguistically-motivated and data-driven approaches both qualitatively and quantitatively.

Automatic Lexical Learning

We have implemented several algorithms for automatic lexical learning in an isolated word recognition setting. An extension to this work would be a dynamic incorporation of these algorithms into dialogue systems. For example, speak-and-spell models, that prompt the user for the spelling of an unrecognized or OOV word, have been successfully implemented within dialogue models for the acquisition of city names [Bauer and Junkawitsch, 1999; Filisko and Seneff, 2005] and proper names [Schramm et al., 2000]. One can envision feeding the spoken instances of the word and its spelling to the Turbo algorithm and dynamically adding the generated lexical entry to the underlying ASR dictionary. It would be interesting to explore the impact of such an approach on the quality of human-machine interaction, particularly when a word that was previously out-of-vocabulary is encountered again in a dialogue.

Improvements and Extensions to the Turbo Algorithm

The Turbo algorithm involved letter and sub-word recognizers that transferred bias information to each other through a bi-directional L2S model. The algorithm implementation entailed a number of parameters associated with the recognizers' *N*-best list size and the weight of the bias. The parameters were tuned in an empirical and local fashion that did not necessarily guarantee a global optimum. Moreover, the parameters were tuned once on a development set. Multiple improvements can be introduced to this approach. For example, the parameters can be optimized simultaneously using, for example, simulated annealing [Kirkpatrick et al., 1983]. The parameters can also be adaptively tuned based on incoming observations.

The core of the Turbo algorithm is the fusion of several sources of information in order to improve overall decoding performance. Such a concept can be extended to different set-ups. For example, a recent approach to unsupervised pattern discovery in speech produced reliable clusters of speech patterns [Park and Glass, 2006]. Such clusters could potentially be mapped to a phonetic representation using sub-word recognition. Since a cluster consists of multiple occurrences of similar acoustic patterns, it can be processed by multiple sub-word recognizers integrated in a feedback structure. Based on the performance observed in this thesis, one can envision an improvement in sub-word recognition performance. On a side note, if sub-word recognition is also followed by S2L, a graphemic representation can be obtained and a lexical entry is learned.

OOV Word Detection

One of the areas that we explored was sub-word modeling for lexical access. A sub-word recognizer was incorporated in parallel with a word recognizer, and was triggered manually whenever the word recognizer failed. An extension of this approach for continuous ASR would be a filler model which would automatically detect the OOV word, and then hypothesize a string of sub-lexical units. The filler approach has been thoroughly investigated with phoneme-based OOV models in [Asadi et al., 1990; Bazzi and Glass, 2000a,b; Bazzi, 2002], and successfully implemented in continuous ASR. The filler model can also consist of sub-syllabic sequences which could yield better performance than phonemes due to the higher linguistic constraint.

OOV Word Modeling

In this thesis, we integrated the sub-word units in a flat hybrid model initially proposed in [Bisani and Ney, 2005] for continuous ASR. The result was a recognizer that could decode a spoken utterance into a string of words and sub-words. The proposed approach can be extended to estimate the graphemic representation of hypothesized OOV words. This can be achieved either by using spellnemes as the basic sub-lexical units or by post-processing sub-word sequences with a S2L model.

Appendix A

The Sub-Lexical Context-Free Grammar

We describe the grammar designed to encode the sub-syllabic knowledge of English words. In particular, we list excerpts of the context-free rules in a hierarchical fashion starting with a root node denoted as `WRD`:

The Second Layer defines the structure of `.WRD` in terms of sub-syllabic units such as `onset` and `rhyme`.

The Third Layer describes how the sub-syllabic structures are pronounced in terms of sub-words that consist of phonemic clusters augmented with positional and phonological knowledge.

The Fourth (Terminal) Layer defines the graphemic representation of the sub-word units.

The following conventions are used for the context-free rules:

a. A term of the form `.<category>` denotes the left-hand side of a context-free rule.

b. The lines following a `.<category>` are alternative right-hand sides and are separated by `||` or by a newline.

c. `[<category>]` denotes optional.

d. `(<category1> <category2> ... <categoryN>)` is equivalent to `<category1> OR <category2> OR ... <categoryN>`.

A.1 The Second Layer: The Sub-Syllabic Representation

```
.WRD
function_word onset rhyme
function_word [usyl]
```

```
rhyme1 (affix usyl affix2)
rhyme1 ambi usyl usyl [affix2]
onset1 rhyme1 [usyl] rhyme (usyl affix2)
onset1 pre rhyme1
[pre] [onset1] rhyme1 usyl [ambi] rhyme [affix]
[onset1] rhyme1 usyl [affix] onset [usyl] rhyme [rhyme]
[onset1] rhyme1 (ambi onset) usyl [affix] rhyme (ambi onset) rhyme
rhyme1 (affix affix2) onset rhyme affix [affix2]
rhyme1 (affix affix2) usyl
rhyme1 [usyl] function_word [rhyme] [affix]
rhyme1 affix [rhyme] function_word onset rhyme
```

A.2 The Third Layer: The Sub-Words

```
.pre
maek (-axl -ax+ -axn -ihng nax+) || -axp || -axb || maxr ||
maxn || maxnt || max+ || maxk || -ihx+ || -uhx+ || -axk ||
-aexl || -ahxv || -aox+ || -ax+ || -axd || -axf || -axg ||
-axjh || -axl || -axm || -axn || -axr || -axs || -axsh ||

.usyl
-axv || shaxn || yaxr || yaxl || -ehxl || -aexl || -aox+ ||
-ihx+ || -iyx+ || -owx+ || -uhx+ || -ax+ || -axb || -axch ||
-axd || -axf || -axg || -axjh || -axk || -axkt || -axl ||
-axld || -axlt || -axm || -axn || -axnch || -axnd || -axnjh ||
-axnt || -axp || -axr || -axrd || -axrg || -axs || -axsh ||

.ambi
de || df || dth || dz || er || g+ || ny || sz || tf || th+

.affix
+jh || +ch || +zh || +sh || +th [+s] || +s +th || +z || +s

.onset
sh+ || sh+ m+ || sht+ || ts+ || kw+ || m+ [y+] || n+ || hh [w+] ||
s+ || w+ || v+ || vr+ || vw+ || k+ (y+ w+) || p+ [y+] || p+ [w+] ||
b+ [w+] || d+ [w+] || d+ || t+ || tw+ || k+ || g+ || y+ || fr+ ||
fy+ || fl+ || dr+ || f+ || gl+ || gr+ || gw+ || jh+ || kl+ ||
kr+ || l+ || bl+ || br+ || ch+ || pl+ || pr+ || r+ || sl+ || sm+ ||
sn+ || sk+ || skw+ || skr+ || sp+ [l+] || spr+ || st+ || str+ ||
th+ [r+] || tr+ || v+ || kl+ || kr+ || zh+ [w+] || z+ [w+] || s+ || w+

.onset1
```

```
fth+ || sth+ || de || s+ || ts+ || s+ f+ || s+ (w+ v+) || dth||
sht+ || kw+  || k+ y+ || m+ y+ || p+ y+ || hh [w+] || s+ [k+] w+||
sh+ (l+ r+ n+ m+ w+ t+) || sh+ || w+ || v+ (l+ r+) || vw+ || v+||
vr+ || p+ [w+] || b+ [w+] || d+ [w+] || t+ || tw+ || k+ [w+] ||
g+ || y+ || fr+ || fl+ || fy+ || dr+ || f+ || gl+ || gr+ || gw+||

   .rhyme1
yaor || yus || yuz || yut || yub || yuk || yum || yun || yu+ ||
-oe+ || -oo+ || -a+ || -aa+ || -aaer || -aaer+ || -aauh+ || -ae+ ||
-aey+ || -ah+ || -ao+ || -aoer+ || -aw+ || -ay+ || -ayiy+ || -eh+ ||
-eher+ || -en+ || -er+ || -ey+ || -eyb || -ih+ || -iy+ || -uh+ ||
-uhng || -uhg || -oy+ || -ow+ || -uw+ || -uwg || -uer+ || -aar ||
-aarsh|| -aer || -ahr || -aor || -aorth || -awr || -ayr || -ehr||
-iehr || -ihr || -owr || -aaen || -ori+ || wuhl || waar || waa+||
-aan || -aarn || -aen || -aern || -ehn || -ahln || -ahlb || -ahn ||

.function_word
of || what || des || the || at || do || who || one || none ||
come || some
```

A.3 The Fourth (Terminal) Layer: The Graphemic Representation

```
.yaor
u [h] r || u r r || e u r e || u r e || u r e

.yu+
e (w u) || i e (w u) || e a u || u [e] || u y ||
y u || u g h || ou || i u || u t

.yut
u t e || u t t e

.yub
u b [e]

.yuk
u q u e || u k e

.yum
u m e
```

```
.yun
y u n e || u n e || y u n

.yus
u s e || e a u c e || o u s

.yuz
u s e

.+s
xx [e] || s [e] || c e || ’ s || z

.+th
t h || t h e

.+zh
g e

.+z
’ s || (s z) || s s || e s || s ’ || z e

.-a+
a

.-aa+
(a o) || a h || a a || a j || a s

.-aab
o (b bb) || a b

.-aach
a ch || o [t] ch || a t ch || a c ch

.-aad
o d || o d d || a d [h]

.-aael
a l

.-aaen
a n

.-aaer
a r
```

```
.-aaer+
a r

.-aaert
a r t

.-aaf
o (f ff) || o p h || a f || a f e || a ff || a a f

.-aag
(a o) g || a g g || o g g || a g u e || o g h

.-aahn
o n

.-aajh
o d g [e] || a g e || a g || a j || a g g

.-aak
o c || e a u c || o (x ck ch c k) || o (ck ch) ||
a (ch c) || a k k || o c c || o ck e || a q || a c ch ||
a c c || a a ck
```

A.4 The Sub-word-to-Phoneme Mapping

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
+ch	ch	+jh	jh
+s	s	+sh	sh
+th	th	+z	z
+zh	zh	-a+	(ax | ey)
-aa+	aa	-aab	aa bd
-aach	aa ch	-aad	aa dd
-aadh	aa dh	-aael	(aa | ae) l
-aaen	(aa | ae) n	-aaer	(aa | ae) r
-aaer+	(aa r | er | ax)	-aaert	(aa r | er | ax) td
-aaf	aa f	-aag	aa gd
-aahn	(aa | ah) n	-aajh	aa jh
-aak	aa kd	-aal	aa l
-aam	aa m	-aamb	aa (m bd | m)
-aamp	aa m pd	-aan	aa n
-aanch	aa n ch	-aand	aa n dd
-aang	aa ng	-aangk	aa ng kd
-aanjh	aa n jh	-aant	aa n td
-aaol	(aa | ao) l	-aap	aa pd
-aar	aa r	-aarb	aa r bd
-aarch	aa r ch	-aard	aa r dd
-aarf	aa r f	-aarg	aa r gd
-aarjh	aa r jh	-aark	aa r kd
-aarl	aa r l	-aarm	aa r m
-aarn	aa r n	-aarp	aa r pd
-aarsh	aa r sh	-aart	aa r td

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-aas	aa s	-aash	aa sh
-aasp	aa s pd	-aast	aa s td
-aat	aa td	-aath	aa th
-aauh+	(aa | uh)	-aav	aa v
-aaxn	(aa | ax) n	-aaz	aa z
-aazh	aa zh	-ae+	ae
-aeb	ae bd	-aech	ae ch
-aed	ae dd	-aedh	ae dh
-aef	ae f	-aeft	ae f td
-aeg	ae gd	-aejh	ae jh
-aek	ae kd	-ael	ae l
-aelb	ae l bd	-aelf	ae l f
-aelp	ae l pd	-aem	ae m
-aemp	ae m pd	-aen	ae n
-aench	ae n ch	-aend	ae n dd
-aeng	ae (ng | ng gd)	-aengk	ae ng kd
-aenjh	ae n jh	-aent	ae n td
-aep	ae pd	-aer	ae r
-aerd	ae r dd	-aerf	(ae | er) f
-aern	(ae | er) n	-aes	ae s
-aesh	ae sh	-aesk	ae s kd
-aesp	ae s pd	-aest	ae s td
-aet	ae td	-aeth	ae th
-aev	ae v	-aexl	(ae | ax) l
-aexn	(ae | ax) n	-aexnd	(ae | ax) n dd
-aexr	(ae | ax) r	-aexs	(ae | ax) s
-aey+	(ae | ey)	-aeyd	(ae | ey) dd
-aez	ae z	-aezh	ae zh
-ah+	ah	-ahb	ah bd
-ahch	ah ch	-ahd	ah dd
-ahdh	ah dh	-ahf	ah f
-ahg	ah g	-ahjh	ah jh
-ahk	ah kd	-ahl	ah l
-ahlb	ah l bd	-ahlf	ah l f
-ahljh	ah l jh	-ahlk	ah l kd
-ahlm	ah l m	-ahln	ah l n
-ahlp	ah l pd	-ahlt	ah l td
-ahm	ah m	-ahmp	ah m pd
-ahn	ah n	-ahnch	ah n ch
-ahnd	ah n dd	-ahng	ah ng
-ahngk	ah ng kd	-ahnjh	ah n jh

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-ahnt	ah n td	-ahp	ah pd
-ahr	ah r	-ahs	ah s
-ahsh	ah sh	-ahsk	ah s kd
-ahst	ah s td	-aht	ah td
-ahth	ah th	-ahv	ah v
-ahxst	(ah | ax) s td	-ahxv	(ah | ax) v
-ahz	ah z	-ao+	ao
-aob	ao bd	-aoch	ao ch
-aod	ao dd	-aoer+	(ao r | er | ax r)
-aoerd	(ao r | er | ax r) dd	-aof	ao f
-aoft	ao f td	-aog	ao gd
-aok	ao kd	-aol	ao l
-aolb	ao l bd	-aold	ao l dd
-aolf	ao l f	-aolk	ao l kd
-aolm	ao l m	-aoln	ao l n
-aolt	ao l td	-aom	ao m
-aomp	ao m pd	-aon	ao n
-aonch	ao n ch	-aong	ao ng
-aongk	ao ng k	-aor	ao r
-aorb	ao r bd	-aorch	ao r ch
-aord	ao r dd	-aorf	ao r f
-aorg	ao r gd	-aorjh	ao r jh
-aork	ao r kd	-aorm	ao r m
-aorn	ao r n	-aorp	ao r pd
-aors	ao r s	-aort	ao r td
-aorth	ao r th	-aos	ao s
-aosh	ao sh	-aost	ao s td
-aot	ao td	-aoth	ao th
-aowl	(ao | ow) l	-aows	(ao | ow) s
-aowt	(ao | ow) td	-aox+	(ao | ax)
-aoxr	(ao | ax) r	-aoz	ao v
-aw+	aw	-awb	aw bd
-awch	aw ch	-awd	aw dd
-awdh	aw dh	-awf	aw f
-awk	aw kd	-awl	aw l
-awlk	aw l kd	-awm	aw m
-awn	aw n	-awnd	aw n dd
-awnt	aw n td	-awr	aw (r | ax r)
-aws	aw s	-awt	aw td

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-awth	aw th	-awv	aw v
-awz	aw z	-ax+	ax
-axb	ax bd	-axch	ax ch
-axd	ax dd	-axf	ax f
-axg	ax gd	-axjh	ax jh
-axk	ax kd	-axkt	ax kd td
-axl	ax l	-axld	ax l dd
-axlt	ax l td	-axm	ax m
-axn	ax n	-axnch	ax n ch
-axnd	ax n dd	-axnjh	ax n jh
-axnt	ax n td	-axp	ax pd
-axr	ax r	-axrd	ax r dd
-axrg	ax r gd	-axs	ax s
-axsh	ax sh	-axsk	ax s kd
-axst	ax s td	-axt	ax td
-axth	ax th	-axv	ax v
-axz	ax z	-axzh	ax zh
-ay+	ay	-ayb	ay bd
-ayd	ay dd	-aydh	ay dh
-ayf	ay f	-ayg	ay gd
-ayiy+	(ay | iy)	-ayiyd	(ay | iy) dd
-ayiyn	(ay | iy) n	-ayjh	ay jh
-ayk	ay kd	-ayl	ay l
-ayld	ay l dd	-aym	ay m
-ayn	ay n	-aynd	ay n dd
-aynt	ay n td	-ayp	ay pd
-ayr	ay r	-ayrm	ay r m
-ayrn	ay r n	-ays	ay s
-aysh	ay sh	-ayst	ay s td
-ayt	ay td	-ayth	ay th
-ayv	ay v	-ayz	ay z
-d	dd	-eh+	eh
-ehb	eh bd	-ehch	eh ch
-ehd	eh dd	-ehdh	eh dh
-eher+	(er | eh r)	-ehf	eh f
-ehg	eh gd	-ehjh	eh jh
-ehk	eh kd	-ehl	eh l
-ehlb	eh l bd	-ehld	eh l dd
-ehlf	eh l f	-ehlg	eh l gd

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-ehlk	eh l kd	-ehlm	eh l m
-ehln	eh l n	-ehlp	eh l pd
-ehlt	eh l td	-ehlv	eh l v
-ehm	eh m	-ehmb	eh m bd
-ehmd	eh m dd	-ehmp	eh m pd
-ehn	eh n	-ehnch	eh n ch
-ehnd	eh n dd	-ehng	eh ng
-ehngk	eh ng kd	-ehnjh	eh n jh
-ehnt	eh n td	-ehp	eh pd
-ehr	(eh , ae) r	-ehrch	eh r ch
-ehrd	eh r dd	-ehrf	eh r f
-ehrn	eh r n	-ehs	eh s
-ehsh	eh sh	-ehsk	eh s kd
-ehst	eh s td	-eht	eh td
-ehth	eh th	-ehv	eh v
-ehxl	(eh | ax) l	-ehxnt	(eh | ax) n td
-ehz	eh z	-ehzh	eh zh
-en+	en	-ent	en td
-enth	en th	-enz	en z
-er+	(er | ax r)	-erb	er bd
-erch	er ch	-erd	er dd
-erdh	er dh	-erf	er f
-erg	er gd	-erjh	er jh
-erk	er kd	-erl	er l
-erld	er l dd	-erm	er m
-ern	er n	-ernd	er n dd
-ernt	er n td	-erp	er pd
-ers	er s	-ersh	er sh
-erst	er s td	-ert	er td
-erth	er th	-erv	er v
-erz	er z	-erzh	er zh
-ey+	ey	-eyb	ey bd
-eych	ey ch	-eyd	ey dd
-eydh	ey dh	-eyf	ey f
-eyg	ey gd	-eyjh	ey jh
-eyk	ey kd	-eyl	ey (ax l | l)
-eym	ey m	-eymb	ey m bd
-eyn	eh n	-eynch	eh n ch

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-eyng	ey ng	-eynjh	ey n jh
-eynt	ey n td	-eyp	ey pd
-eys	ey s	-eysh	ey sh
-eysk	ey s kd	-eyst	ey s td
-eyt	ey td	-eyth	ey th | ey th
-eyv	ey v	-eyz	ey z
-eyzh	ey zh	-f	f
-iehr	(ih | eh) r	-ih+	ih
-ihb	ih bd	-ihch	(ih | ax) ch
-ihd	ih dd	-ihdh	ih dh
-ihers	(ih r | er) s	-ihf	ih f
-ihft	ih f td	-ihg	ih gd
-ihjh	ih jh	-ihk	ih kd
-ihl	ih l	-ihlb	ih l bd
-ihld	ih l dd	-ihlf	ih l f
-ihlg	ih l gd	-ihlk	ih l kd
-ihln	ih l n	-ihlt	ih l td
-ihm	ih m	-ihmb	ih m bd
-ihmp	ih m pd	-ihn	ih n
-ihnch	ih n ch	-ihnd	ih n dd
-ihng	ih (ng | ng gd)	-ihngk	ih ng kd
-ihnjh	ih n jh	-ihnt	ih n td
-ihp	ih pd	-ihr	ih r
-ihrd	ih r dd	-ihs	ih s
-ihsh	ih sh	-ihsk	ih s kd
-ihsp	ih s pd	-iht	ih td
-ihth	ih th	-ihv	(ih | ax) v
-ihx+	(ih | ax)	-ihxk	(ih | ax) kd
-ihxl	(ih | ax) l	-ihxn	(ih | ax) n
-ihxs	(ih | ax) s	-ihxt	(ih | ax) td
-ihz	ih z	-ihzh	ih zh
-is	(ax s | iy)	-iy+	iy
-iyb	iy bd	-iych	iy ch
-iyd	iy dd	-iydh	iy dh
-iyf	iy f	-iyg	iy gd
-iyk	iy kd	-iyl	iy l
-iyld	iy l dd	-iym	iy m
-iyn	iy n	-iyp	iy pd

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-iys	iy s	-iysh	iy sh
-iyst	iy s td	-iyt	iy td
-iyth	iy th	-iyv	iy v
-iyx+	(iy | ax)	-iyxl	(iy | ax) l
-iyxm	(iy | ax) m	-iyxn	(iy | ax) n
-iyxs	(iy | ax) s	-iyxv	(iy | ax) v
-iyxz	(iy | ax) z	-iyz	iy z
-iyzh	iy zh	-k	kd
-m	(m | ax m)	-n	(n | ax n)
-oe+	(ih | uh)	-oo+	(ow | uw)
-ori+	(ao r ax | ao r)	-ow+	ow
-owb	ow bd	-owch	ow ch
-owd	ow dd	-owdh	ow dh
-owf	ow f	-owft	ow f td
-owg	ow gd	-owjh	ow jh
-owk	ow kd	-owl	ow l
-owlb	ow l bd	-owld	ow l dd
-owlf	ow l f	-owlm	ow l m
-owln	ow l n	-owlp	ow l pd
-owlt	ow l td	-owm	ow m
-own	ow n	-ownt	ow n td
-owp	ow pd	-owr	ow r
-ows	ow s	-owsh	ow sh
-owst	ow s td	-owt	ow td
-owth	ow th	-owuhk	(ow | uh) kd
-owv	ow v	-owx+	(ow | ax)
-owxl	(ow | ax) l	-owxm	(ow | ax) m
-owxz	(ow | ax) z	-owz	ow z
-owzh	ow zh	-oy+	oy
-oyd	oy dd	-oyg	oy gd
-oyl	oy l	-oyn	oy n
-oynt	oy n td	-oys	oy s
-oyt	oy td	-oyth	oy th
-oyz	oy z	-p	pd
-pt	pd td	-sk	s kd
-st	s td	-t	td
-ts	td s | td s	-uer+	(w eh | ao) r
-uh+	uh	-uhd	uh dd

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
-uhf	uh f	-uhg	uh gd
-uhk	uh kd	-uhl	uh l
-uhlf	uh l f	-uhlp	uh l pd
-uhlt	uh l td	-uhm	uh m
-uhn	uh n	-uhng	uh ng
-uhs	uh s	-uhsh	uh sh
-uht	uh td	-uhx+	(ax | uw)
-uw+	uw	-uwb	uw bd
-uwch	uw ch	-uwd	uw dd
-uwdh	uw dh	-uwf	uw f
-uwg	uw gd	-uwjh	uw jh
-uwk	uw kd	-uwl	uw l
-uwld	uw l dd	-uwlk	uw l kd
-uwm	uw m	-uwn	uw n
-uwng	uw ng	-uwnt	uw n td
-uwp	uw pd	-uws	uw s
-uwsh	uw sh	-uwst	uw s td
-uwt	uw td	-uwth	uw th
-uwv	uw v	-uwz	uw z
-uwzh	uw zh	-v	v
a	(ax | ey)	and	(ax | ae) n dd
are	aa r	as	(ax | ae) z
at	(ae | ax) td	b+	b
bl+	b l	br+	b r
ch+	ch	come	k ah m
d+	d	de	d (eh | ey | iy)
des	d ey	df	df
do	d uw	dr+	dr r
dth	(dh | th)	dz	(dd z | z)
er	er	f+	f
fl+	f l	fr+	f r
from	f r (ah | ax) m	fth+	(f th | th)
fy+	f y	g+	g
gl+	g l	gr+	g r
gw+	g w	ham	(hh ae | hh ax | ax) m
has	hh ae (s | z) | hh ae z	herst	(hh er | er) s td
hh	hh	i	ay
it	ih td	jh+	jh

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
k+	k	kl+	k l
kr+	k r	kw+	k w
l+	l	lives	l (ay | ih) v z
los	l (ow | ao | aa) s	m+	m
maek	m ae kd	max+	m ax
maxg	m ax gd	maxjh	m ax jh
maxk	m ax kd	maxl	m ax l
maxm	m ax m	maxn	m ax n
maxnd	m ax n dd	maxnt	m ax n td
maxr	m ax r	maxth	m ax th
maxz	m ax z	maxzh	m ax zh
me	m iy	n+	n
nax+	n ax	naxk	n ax kd
naxl	n ax l	naxm	n ax m
naxn	n ax n	naxs	n ax s
naxt	n ax td	naxv	n ax v
none	n ah n	not	n aa td
ny	n y	of	ax v
on	(ah | aa) n	one	w ah n
our	(aw er | aa r)	p+	p
pl+	p l	pr+	p r
r+	r	rax+	(r | r ax)
rsyl	(r | ax r | er)	s+	s
sh+	sh	shax	(sh ax | sh)
shaxn	sh ax n	sht+	sh t
sk+	s k-	skr+	s k- r
skw+	s k- w	sl+	s l
sm+	s m	sn+	s n
some	s ah m	sp+	s p-
spr+	s p- r	st+	s t-
sth+	s th	str+	s tr r
sz	(z | s)	t+	t
tf	tf	th+	th
that	dh ae td	the	dh (ax | ah | ih | iy)
then	dh eh n	to	tf (ax | uw)
tqen+	tq en	tqens	tq en s
tqent	tq en td	tr+	tr r
ts+	td s	tw+	t w

Sub-word	Phonemic Pronunciation	Sub-word	Phonemic Pronunciation
ugh	(ax | ow | gd)	us	ah s
v+	v	vr+	v r
vw+	v w	w+	w
waa+	w aa	waar	w aa r
was	w ah z	what	w (ax | ah) td
when	w eh n	who	hh (uw | uw m | uw z)
wuhl	w uh l	y+	y
yaor	y ao r	yaxl	y ax l
yaxr	y ax r	yeah	y (ae | eh | ey ax)
you	y (uw | ax)	your	(y uw r | y ao r | y er)
yu+	y uw	yub	y uw bd
yuk	y uw kd	yum	y uw m
yun	y uw n	yus	y uw s
yut	y uw td	yuz	y uw z
yy	(y | iy)	z+	z
zh+	zh		

Table A.1: The linguistically-motivated sub-word units and their corresponding phonemic representation.

Appendix B

The Phonetic Alphabet

IPA	ARPA	Example	IPA	ARPA	Example	IPA	ARPA	Example
[ɑ]	aa	b*o*b	[ɨ]	ix	deb*i*t	[ɪ]	ih	b*i*t
[æ]	ae	b*a*t	[i]	iy	b*ee*t	[y]	y	*y*acht
[ʌ]	ah	b*u*t	[ɑ^{w}]	aw	b*ou*t	[ɔ]	ao	b*ou*ght
[ɛ]	eh	b*e*t	[ɝ]	er	b*ir*d	[e]	ey	b*ai*t
[ə]	ax	*a*bout	[ə^{h}]	ax-h	p*o*tato	[ɚ]	axr	butt*er*
[ɑ^{y}]	ay	b*i*te	[u]	uw	b*oo*t	[ʊ]	uh	b*oo*k
[ü]	ux	t*oo*t	[o]	ow	b*oa*t	[ɔ^{y}]	oy	b*oy*
[m]	m	*mom*	[n]	n	*noon*	[ŋ]	ng	si*ng*
[l]	l	*l*ay	[b]	b	*b*ee	[b$^{\square}$]	bcl	b closure
[ɾ̃]	nx	wi*nn*er	[ŋ̩]	eng	Wash*ing*ton	[▯]	epi	epenthetic silence
[d]	d	*d*ay	[d$^{\square}$]	dcl	d closure	[ɾ]	dx	mu*dd*y
[p]	p	*p*ea	[p$^{\square}$]	pcl	p closure	[ʔ]	q	glottal stop
[l̩]	el	bott*le*	[m̩]	em	bott*om*	[n̩]	en	butt*on*
[ǰ]	jh	*j*oke	[k]	k	*k*ey	[k$^{\square}$]	kcl	k closure
[s]	s	*s*ea	[š]	sh	*sh*e	[č]	ch	*ch*oke
[t]	t	*t*ea	[t$^{\square}$]	tcl	t closure	[θ]	th	*th*in
[r]	r	*r*ay	[f]	f	*f*in	[v]	v	*v*an
[w]	w	*w*ay	[ɡ]	g	*g*ay	[ɡ$^{\square}$]	gcl	g closure
[h]	hh	*h*ay	[ɦ]	hv	a*h*ead	[□]	pau	pause
[ð]	dh	*th*en	[z]	z	*z*one	[ž]	zh	a*z*ure

Table B.1: IPA and ARPAbet symbols for the phones in the English language with sample occurrences.

Appendix C

Rhyme Splitting

We illustrate the splitting of the sub-words corresponding to the rhyme structure into a nucleus and a coda. If a sub-word consists only of a vowel sound, e.g. `-aauh+`, it is not further split, since it, originally, consisted of a nucleus only.

Rhyme	Split Rhyme	Rhyme	Split Rhyme	Rhyme	Split Rhyme
yaor	yao! !r	yus	yu! !s	yuz	yu! !z
yut	yu! !t	yub	yu! !b	yuk	yu! !k
yum	yu! !m	yun	yu! !n	yu+	yu+
-oe+	-oe+	-oo+	-oo+	-a+	-a+
-aa+	-aa+	-aaer	-aaer	-aaer+	-aaer+
-aauh+	-aauh+	-ae+	-ae+	-aey+	-aey+
-ah+	-ah+	-ao+	-ao+	-aoer+	-aoer+
-aw+	-aw+	-ay+	-ay+	-ayiy+	-ayiy+
-eh+	-eh+	-eher+	-eher+	-en+	-en+
-er+	-er+	-ey+	-ey+	-eyb	-ey! !b
-ih+	-ih+	-iy+	-iy+	-uh+	-uh+
-uhng	-uh! !ng	-uhg	-uh! !g	-oy+	-oy+
-ow+	-ow+	-uw+	-uw+	-uwg	-uw! !g
-uer+	-uer+	-aar	-aa! !r	-aarsh	-aa! !rsh
-aer	-aer	-ahr	-ah! !r	-aor	-ao! !r
-aorth	-ao! !rth	-awr	-awr	-ayr	-ay! !r
-ehr	-eh! !r	-iehr	-ieh! !r	-ihr	-ih! !r
-owr	-ow! !r	-aaen	-aae! !n	-ori+	-ori+
wuhl	wuh! !l	waar	waa! !r	waa+	waa+
-aan	-aa! !n	-aarn	-aa! !rn	-aen	-ae! !n
-aern	-aer! !n	-ehn	-eh! !n	-ahln	-ah! !ln
-ahlb	-ah! !lb	-ahn	-ah! !n	-aoln	-ao! !ln
-aon	-ao! !n	-aorn	-ao! !rn	-awn	-aw! !n
-ayiyn	-ayiy! !n	-ayn	-ay! !n	-ayrn	-ay! !rn
-ehln	-eh! !ln	-ehlg	-eh! !lg	-ehlv	-eh! !lv
-ehrn	-eh! !rn	-ern	-er! !n	-eyn	-ey! !n
-ihln	-ih! !ln	-ihn	-ih! !n	-iyn	-iy! !n

Rhyme	Split Rhyme	Rhyme	Split Rhyme	Rhyme	Split Rhyme
-owln	-ow! !ln	-own	-ow! !n	-oyn	-oy! !n
-uhn	-uh! !n	-uwn	-uw! !n	-aem	-ae! !m
-aarm	-aa! !rm	-aam	-aa! !m	-ahlm	-ah! !lm
-ahm	-ah! !m	-aolm	-ao! !lm	-aom	-ao! !m
-aorm	-ao! !rm	-awm	-aw! !m	-aym	-ay! !m
-ayrm	-ay! !rm	-ehlm	-eh! !lm	-ehm	-eh! !m
-erm	-er! !m	-eym	-ey! !m	-ihm	-ih! !m
-iym	-iy! !m	-owlm	-ow! !lm	-owm	-ow! !m
-uhm	-uh! !m	-uwm	-uw! !m	-aael	-aae! !l
-aal	-aa! !l	-aaol	-aao! !l	-aarl	-aa! !rl
-ael	-ae! !l	-ahl	-ah! !l	-aol	-ao! !l
-aowl	-aow! !l	-awl	-aw! !l	-ayl	-ay! !l
-ehl	-eh! !l	-erl	-er! !l	-eyl	-ey! !l
-ihl	-ih! !l	-iyl	-iy! !l	-owl	-ow! !l
-oyl	-oy! !l	-uhl	-uh! !l	-uwl	-uw! !l
-aab	-aa! !b	-aach	-aa! !ch	-aad	-aa! !d
-aadh	-aa! !dh	-aaert	-aaer! !t	-aaf	-aa! !f
-aag	-aa! !g	-aajh	-aa! !jh	-aak	-aa! !k
-aamb	-aa! !mb	-aamp	-aa! !mp	-aanch	-aa! !nch
-aand	-aa! !nd	-aang	-aa! !ng	-aangk	-aa! !ngk
-aanjh	-aa! !njh	-aant	-aa! !nt	-aap	-aa! !p
-aarb	-aa! !rb	-aarch	-aa! !rch	-aard	-aa! !rd
-aarf	-aa! !rf	-aarg	-aa! !rg	-aarjh	-aa! !rjh
-aark	-aa! !rk	-aarp	-aa! !rp	-aart	-aa! !rt
-aas	-aa! !s	-aash	-aa! !sh	-aasp	-aa! !sp
-aast	-aa! !st	-aat	-aa! !t	-aath	-aa! !th
-aav	-aa! !v	-aaz	-aa! !z	-aazh	-aa! !zh
-aeb	-ae! !b	-aech	-ae! !ch	-aed	-ae! !d
-aedh	-ae! !dh	-aef	-ae! !f	-aeft	-ae! !ft
-aeg	-ae! !g	-aejh	-ae! !jh	-aek	-ae! !k
-aelb	-ae! !lb	-aelf	-ae! !lf	-aelp	-ae! !lp
-aemp	-ae! !mp	-aench	-ae! !nch	-aend	-ae! !nd
-aeng	-ae! !ng	-aengk	-ae! !ngk	-aenjh	-ae! !njh
-aent	-ae! !nt	-aep	-ae! !p	-aerd	-aer! !d
-aerf	-aer! !f	-aes	-ae! !s	-aesp	-ae! !sp
-aesh	-ae! !sh	-aesk	-ae! !sk	-aest	-ae! !st
-aet	-ae! !t	-aeth	-ae! !th	-aev	-ae! !v
-aeyd	-aey! !d	-aez	-ae! !z	-aezh	-ae! !zh
-ahb	-ah! !b	-ahch	-ah! !ch	-ahd	-ah! !d

Rhyme	Split Rhyme	Rhyme	Split Rhyme	Rhyme	Split Rhyme
-ahdh	-ah! !dh	-ahf	-ah! !f	-ahg	-ah! !g
-ahjh	-ah! !jh	-ahk	-ah! !k	-ahlf	-ah! !lf
-ahlk	-ah! !lk	-ahlp	-ah! !lp	-ahlt	-ah! !lt
-ahljh	-ah! !ljh	-ahmp	-ah! !mp	-ahnch	-ah! !nch
-ahnd	-ah! !nd	-ahng	-ah! !ng	-ahngk	-ah! !ngk
-ahnjh	-ah! !njh	-ahnt	-ah! !nt	-ahp	-ah! !p
-ahs	-ah! !s	-ahsh	-ah! !sh	-ahsk	-ah! !sk
-ahst	-ah! !st	-aht	-ah! !t	-ahth	-ah! !th
-ahv	-ah! !v	-ahz	-ah! !z	-aob	-ao! !b
-awb	-aw! !b	-aoch	-ao! !ch	-aod	-ao! !d
-aoerd	-aoer! !d	-aof	-ao! !f	-aoft	-ao! !ft
-aog	-ao! !g	-aok	-ao! !k	-aolb	-ao! !lb
-aold	-ao! !ld	-aolf	-ao! !lf	-aolk	-ao! !lk
-aolt	-ao! !lt	-aomp	-ao! !mp	-aonch	-ao! !nch
-aong	-ao! !ng	-aongk	-ao! !ngk	-aorb	-ao! !rb
-aors	-ao! !rs	-aorch	-ao! !rch	-aord	-ao! !rd
-aorf	-ao! !rf	-aorg	-ao! !rg	-aorjh	-ao! !rjh
-aork	-ao! !rk	-aorp	-ao! !rp	-aort	-ao! !rt
-aos	-ao! !s	-aosh	-ao! !sh	-aost	-ao! !st
-aot	-ao! !t	-aoth	-ao! !th	-aows	-aow! !s
-aowt	-aow! !t	-aoz	-ao! !z	-awch	-aw! !ch
-awd	-aw! !d	-awdh	-aw! !dh	-awf	-aw! !f
-awk	-aw! !k	-awlk	-aw! !lk	-awnd	-aw! !nd
-awnt	-aw! !nt	-aws	-aw! !s	-awt	-aw! !t
-awth	-aw! !th	-awv	-aw! !v	-awz	-aw! !z
-ayb	-ay! !b	-ayd	-ay! !d	-aydh	-ay! !dh
-ayf	-ay! !f	-ayg	-ay! !g	-ayiyd	-ayiy! !d
-ayjh	-ay! !jh	-ayk	-ay! !k	-ayld	-ay! !ld
-aynd	-ay! !nd	-aynt	-ay! !nt	-ayp	-ay! !p
-ays	-ay! !s	-ayst	-ay! !st	-aysh	-ay! !sh
-ayt	-ay! !t	-ayth	-ay! !th	-ayv	-ay! !v
-ayz	-ay! !z	-ehb	-eh! !b	-ehch	-eh! !ch
-ehd	-eh! !d	-ehdh	-eh! !dh	-ehf	-eh! !f
-ehg	-eh! !g	-ehjh	-eh! !jh	-ehk	-eh! !k
-ehlb	-eh! !lb	-ehld	-eh! !ld	-ehlf	-eh! !lf
-ehlk	-eh! !lk	-ehlp	-eh! !lp	-ehlt	-eh! !lt
-ehmb	-eh! !mb	-ehmd	-eh! !md	-ehmp	-eh! !mp
-ehnch	-eh! !nch	-ehnd	-eh! !nd	-ehng	-eh! !ng
-ehngk	-eh! !ngk	-ehnjh	-eh! !njh	-ehnt	-eh! !nt
-ehp	-eh! !p	-ehrch	-eh! !rch	-ehrd	-eh! !rd

Rhyme	Split Rhyme	Rhyme	Split Rhyme	Rhyme	Split Rhyme
-ehrf	-eh! !rf	-ehs	-eh! !s	-ehsh	-eh! !sh
-ehsk	-eh! !sk	-ehst	-eh! !st	-eht	-eh! !t
-ehth	-eh! !th	-ehv	-eh! !v	-ehz	-eh! !z
-ehzh	-eh! !zh	-ent	-en! !t	-enth	-en! !th
-enz	-en! !z	-erb	-er! !b	-erch	-er! !ch
-erd	-er! !d	-erdh	-er! !dh	-erf	-er! !f
-erg	-er! !g	-erjh	-er! !jh	-erk	-er! !k
-erld	-er! !ld	-ernd	-er! !nd	-ernt	-er! !nt
-erp	-er! !p	-ers	-er! !s	-ersh	-er! !sh
-erst	-er! !st	-ert	-er! !t	-erth	-er! !th
-erv	-er! !v	-erz	-er! !z	-erzh	-er! !zh
-eych	-ey! !ch	-eyd	-ey! !d	-eyf	-ey! !f
-eyg	-ey! !g	-eyjh	-ey! !jh	-eyk	-ey! !k
-eymb	-ey! !mb	-eynch	-ey! !nch	-eyng	-ey! !ng
-eynjh	-ey! !njh	-eynt	-ey! !nt	-eyp	-ey! !p
-eys	-ey! !s	-eysh	-ey! !sh	-eysk	-ey! !sk
-eyst	-ey! !st	-eyt	-ey! !t	-eyth	-ey! !th
-eydh	-ey! !dh	-eyv	-ey! !v	-eyz	-ey! !z
-eyzh	-ey! !zh	-ihb	-ih! !b	-ihch	-ih! !ch
-ihd	-ih! !d	-ihdh	-ih! !dh	-ihers	-iher! !s
-ihf	-ih! !f	-ihft	-ih! !ft	-ihg	-ih! !g
-ihjh	-ih! !jh	-ihk	-ih! !k	-ihlb	-ih! !lb
-ihld	-ih! !ld	-ihlf	-ih! !lf	-ihlg	-ih! !lg
-ihlk	-ih! !lk	-ihlt	-ih! !lt	-ihmb	-ih! !mb
-ihmp	-ih! !mp	-ihnch	-ih! !nch	-ihnd	-ih! !nd
-ihng	-ih! !ng	-ihngk	-ih! !ngk	-ihnjh	-ih! !njh
-ihnt	-ih! !nt	-ihp	-ih! !p	-ihrd	-ih! !rd
-ihs	-ih! !s	-ihsh	-ih! !sh	-ihsk	-ih! !sk
-ihsp	-ih! !sp	-iht	-ih! !t	-ihth	-ih! !th
-ihv	-ih! !v	-ihz	-ih! !z	-ihzh	-ih! !zh
-is	-is	-iyb	-iy! !b	-iych	-iy! !ch
-iyd	-iy! !d	-iydh	-iy! !dh	-iyf	-iy! !f
-iyg	-iy! !g	-iyk	-iy! !k	-iyld	-iy! !ld
-iyp	-iy! !p	-iys	-iy! !s	-iysh	-iy! !sh
-iyst	-iy! !st	-iyt	-iy! !t	-iyth	-iy! !th
-iyv	-iy! !v	-iyz	-iy! !z	-iyzh	-iy! !zh
-owb	-ow! !b	-owch	-ow! !ch	-owd	-ow! !d
-owdh	-ow! !dh	-owf	-ow! !f	-owft	-ow! !ft
-owg	-ow! !g	-owjh	-ow! !jh	-owk	-ow! !k
-owlb	-ow! !lb	-owld	-ow! !ld	-owlf	-ow! !lf

Rhyme	Split Rhyme	Rhyme	Split Rhyme	Rhyme	Split Rhyme
-owlp	-ow! !lp	-owlt	-ow! !lt	-ownt	-ow! !nt
-owp	-ow! !p	-ows	-ow! !s	-owsh	-ow! !sh
-owst	-ow! !st	-owt	-ow! !t	-owth	-ow! !th
-owuhk	-owuh! !k	-owv	-ow! !v	-owz	-ow! !z
-owzh	-ow! !zh	-oyd	-oy! !d	-oyg	-oy! !g
-oynt	-oy! !nt	-oys	-oy! !s	-oyt	-oy! !t
-oyth	-oy! !th	-oyz	-oy! !z	-uhd	-uh! !d
-uhf	-uh! !f	-uhk	-uh! !k	-uhlf	-uh! !lf
-uhlp	-uh! !lp	-uhlt	-uh! !lt	-uhs	-uh! !s
-uhsh	-uh! !sh	-uht	-uh! !t	-uwb	-uw! !b
-uwch	-uw! !ch	-uwd	-uw! !d	-uwdh	-uw! !dh
-uwf	-uw! !f	-uwjh	-uw! !jh	-uwk	-uw! !k
-uwld	-uw! !ld	-uwlk	-uw! !lk	-uwng	-uw! !ng
-uwnt	-uw! !nt	-uwp	-uw! !p	-uws	-uw! !s
-uwsh	-uw! !sh	-uwst	-uw! !st	-uwt	-uw! !t
-uwth	-uw! !th	-uwv	-uw! !v	-uwz	-uw! !z
-uwzh	-uw! !zh				

Table C.1: The total number of rhymes in the sub-word units is 487, and most are split into nucleus and coda (if possible). The *!* at the end and beginning of each unit denote the nucleus and coda respectively. If a rhyme ends with the diacritic +, then it corresponds to a vowel sound and is itself a nucleus, so it is not split any further.

Appendix D

Sample Queries

A sample bigram query corresponding to the top-10 hypotheses displayed in Table D.1. Bigram terms are combined with each other with implicit `OR`s.

10-best list
she had something `br+ -ey+ k+ -ax+ b+ -axl`
is she had something `br+ -ey+ k+ -ax+ b+ -axl`
as she had something `br+ -ey+ k+ -ax+ b+ -axl`
she had something `br+ -ey+ k+ -ax+ b+ -axl +z`
she had something `br+ -ey+ k+ -ax+ b+ -axl -d`
she had something `br+ -ey+ k+ -ax+ b+ -axl -iy+`
she had something `br+ -ax+ k+ -ax+ b+ -axl`
she had something `br+ -ey+ k+ -er+ b+ -axl`
she had something `br+ -eh+ k+ -ax+ b+ -axl`
verse you had something `br+ -ey+ k+ -ax+ b+ -axl`

Table D.1: The 10-best output of a hybrid recognizer with a 3% OOV rate for the utterance *“she had something breakable”*.

```
 ("she had" "had something" "something br+" "br+ -ey+"
        "-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
                            OR
  ("is she" "she had" "had something" "something br+"
  "br+ -ey+" "-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
                            OR
  ("as she" "she had" "had something" "something br+"
  "br+ -ey+" "-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
                            OR
("she had" "had something" "something br+"  "br+ -ey+"
  "-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl" "-axl +z")
                            OR
```

```
("she had" "had something" "something br+" "br+ -ey+"
"-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl" "-axl -d")
OR
("she had" "had something" "something br+" "br+ -ey+"\verb
"-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl" "-axl -iy+")
OR
("she had" "had something" "something br+" "br+ -ax+"
"-ax+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
OR
("she had" "had something" "something br+" "br+ -ey+"
"-ey+ k+" "k+ -er+" "-er+ b+" "b+ -axl")
OR
("she had" "had something" "something br+" "br+ -eh+"
"-eh+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
OR
("verse you" "you had" "had something" "something br+"
"br+ -ey+" "-ey+ k+" "k+ -ax+" "-ax+ b+" "b+ -axl")
```

A sample bigram query corresponding to the confusion network in Figure D.2.

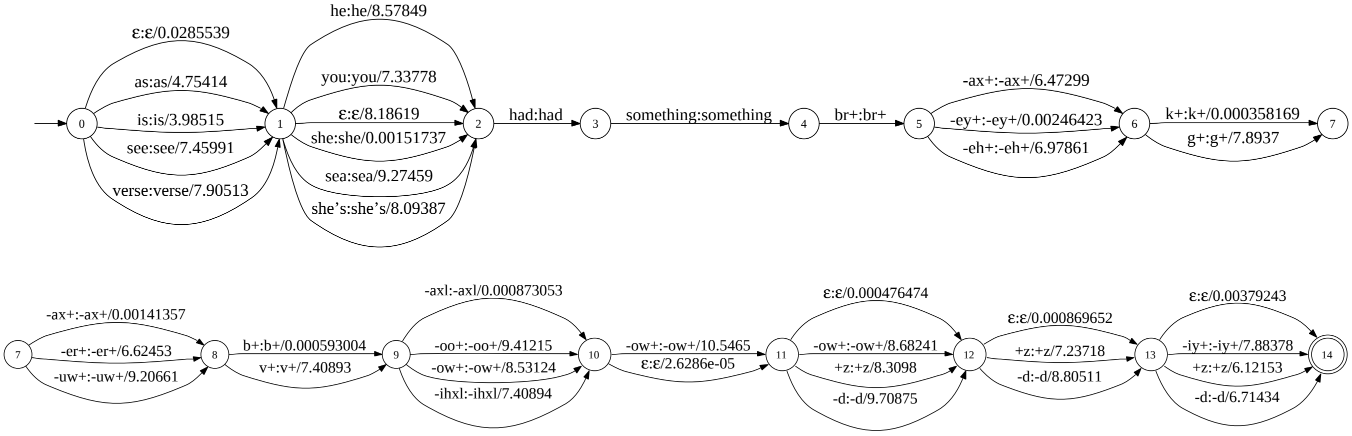


Table D.2: The confusion network generated by a hybrid recognizer with a 3% OOV rate for the utterance *"she had something breakable"*. The network figure is split in half for lack of space and is read left to right. Note that the confusion network is inclusive of the 10-best list shown in Table 7.4.

```
( "as she" "as you" "as he" "as she's" "as sea" "as"
"verse she" "verse you" "verse he" "verse she's"
"verse sea" "verse" "she" "you" "he" "she's" "sea"
"is she" "is you" "is he" "is she's" "is sea" "is"
"see she" "see you" "see he" "see she's" "see sea "see"
OR
```

```
     ( "she had" "you had" "he had" "she’s had" "sea had" "had")
                              OR
                       ("had something")
                              OR
                      ( "something br+")
                              OR
             ( "br+ -eh+" "br+ -ax+" "br+ -ey+")
                              OR
 ( "-eh+ k+" "-eh+ g+" "-ax+ k+" "-ax+ g+" "-ey + k+" "-ey+ g+
                              OR
 ( "k+ -uw+" "k+ -er+" "k+ -ax+" "g+ -uw+" "g+ -er+" "g+ -ax+")
                              OR
 ( "-uw+ b+" "-uw+ v+" "-er+ b+" "-er+ v+" "-ax+ b+" "-ax+ v+")
                              OR
 ( "b+ -axl" "b+ -ow+" "b+ -ihxl" "b+ -oo+" "v+ -axl" "v+ -ow+"
                    "v+ -ihxl" "v+ -oo+")
                              OR
   ( "-axl -ow+"  "-axl "  "-ow+ -ow+"  "-ow+ "  "-ihxl -ow+"
                 "-ihxl "  "-oo+ -ow+"  "-oo+ ")
                              OR
   ( "-ow+ +z" "-ow+ -ow+" "-ow+" "-ow+ -d" "+z" "-ow+" "-d")
                              OR
    ( "+z +z" "+z " "+z -d" "-ow+ +z" "-ow+" "-ow+ -d" "+z"
                  "-d" "-d +z" "-d " "-d -d" )
                              OR
  ( "+z -iy+" "+z +z" "+z " "+z -d" "-iy+" "+z" "-d" "-d -iy+"
                    "-d +z" "-d "  "-d -d")
```

The parentheses “()” allow the grouping of terms. Bigram terms are combined with each other with implicit ORs.

Bibliography

F. Alleva and K. F. Lee. Automatic new word acquisition: spelling from acoustics. In *Proc. of the DARPA Speech and Natural Language Workshop*, pages 266–270, Harwichport, MA, October 1989.

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

A. Asadi and H. C. Leung. New-word addition and adaptation in a stochastic explicit-segment speech recognition system. In *Proc. ICASSP '93*, pages 642–645, Minneapolis, MN, April 1993.

A. Asadi, R. Schwartz, and J. Makhoul. Automatic detection of new words in a large vocabulary continuous speech recognition system. In *Proc. ICASSP '90*, pages 125–128, Albuquerque, NM, April 1990.

A. Asadi, R. Schwartz, and J. Makhoul. Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system. In *Proc. ICASSP '91*, pages 305–308, Toronto, Canada, 1991.

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

L. R. Bahl, S. Das, P. V. de Souza, M. Epstein, R. L. Mercer, B. Merialdo, D. Nahamo, M. A. Picheny, and J. Powell. Automatic phonetic baseform determination. In *Proc. ICASSP '91*, Toronto, Canada, May 1991.

J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S. W. Kuo. Experiments in spoken queries for document retrieval. In *Proc. European Conf. on Speech Communication and Technology*, pages 1323–1326, Rodhes, Greece, September 2007.

C. D. Bartels and J. A. Bilmes. Use of syllable nuclei locations to improve ASR. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 335–340, Kyoto, Japan, December 2008.

J. G. Bauer and J. Junkawitsch. Accurate recognition of city names with spelling as a fall back strategy. In *Proc. European Conf. on Speech Communication and Technology*, pages 263–266, Budapest, Hungary, September 1999.

L. E. Baum. An inequality and associated maximization technique in statistical estimation of a Markov process. *Inequalities*, 3(1):1–8, 1972.

I. Bazzi. *"Modelling Out-of-Vocabulary Words for Robust Speech Recognition"*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 2002.

I. Bazzi and J. R. Glass. Modelling out-of-vocabulary words for robust speech recognition. In *Proc. ICSLP '00*, pages 401–404, Beijing, China, October 2000a.

I. Bazzi and J. R. Glass. Heterogeneous lexical units for automatic speech recognition: Preliminary investigations. In *Proc. ICASSP '00*, pages 1257–1260, Istanbul, Turkey, June 2000b.

I. Bazzi and J. R. Glass. Learning units for domain-independent out-of-vocabulary word modeling. In *Proc. European Conf. on Speech Communication and Technology*, pages 61–64, Aalborg, Denmark, 2001.

I. Bazzi and J. R. Glass. A multi-class approach for modelling out-of-vocabulary words. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 1613–1616, Denver, CO, September 2002.

C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-Codes. In *Proc. ICC*, pages 1064–1070, Geneva, Switzerland, 1993.

M. Bisani and H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 105–108, Denver, CO, September 2002.

M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Proc. Interspeech*, pages 725–728, Lisbon, Portugal, September 2005.

M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

A. W. Black, K. Lenzo, and V. Pagel. Issues in building general letter to sound rules. In *Proc. of the 3rd ESCA Workshop on Speech Synthesis*, pages 77–80, Jenolan Caves, Australia, November 1998.

L. Burnard. British National Corpus: User's Reference Guide for the British National Corpus. *Oxford University Computing Service*, pages 13–19, 1995.

E. Chang, F. Seide, H. M. Meng, Z. Cheng, Y. Shi, and Y. Li. A system for spoken query information retrieval on mobile devices. *IEEE Trans. on Speech and Audio Proc.*, 10(8):531–541, 2002.

J. Chang. *"Near-miss modeling: a segment-based approach to speech recognition"*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1998.

J. Chang and J. R. Glass. Segmentation and modeling in segment-based recognition. In *Proc. European Conf. on Speech Communication and Technology*, pages 1199–1202, Rhodes, Greece, October 1997.

E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proc. National Conf. Artificial Intelligence*, pages 598–603, Providence, RI, July 1997. AAAI Press/MIT Press.

S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig. Advances in Speech Transcription at IBM under the DARPA EARS Program. *IEEE Trans. Speech and Audio Processing*, 14(5):1596–1608, 2006.

S. F. Chen. Conditional and joint models for grapheme-to-phoneme conversion. In *Proc. European Conf. on Speech Communication and Technology*, pages 2033–2036, Geneva, Switzerland, September 2003.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of the ACL*, pages 310–318, Santa Cruz, CA, June 1996.

N. Chomsky and M. Halle. *The sound pattern of English.* Harper and Row, New York, NY, 1968.

G. F. Choueiter, S. Seneff, and J. R. Glass. New word acquisition using subword modeling. In *Proc. Interspeech*, pages 1765–1768, Antwerp, Belgium, August 2007.

Y. L. Chow, M. O. Dunham nd O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos, and R. M. Schwartz. BYBLOS: The BBN continuous speech recognition system. In *Proc. ICASSP '87*, pages 89–92, Dallas, TX, April 1987.

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

G. Chung. A three-stage solution for flexible vocabulary speech understanding. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 266–269, Beijing, China, October 2000a.

G. Chung. Automatically incorporating unknown words in JUPITER. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 520–523, Beijing, China, October 2000b.

G. Chung, C. Wang, S. Seneff, E. Filisko, and M. Tang. Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation. In *Proc. Interspeech*, pages 328–332, Jeju, South Korea, October 2004.

K. W. Church. *"Phrase-Structuring Parsing: A method for taking advantage of allophonic constraints"*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, January 1983.

G. N. Clements and S. J. Keyser. *CV Phonology, A generative theory of the syllable.* Linguistic Inquiry, Cambridge, MA, 1983.

F. Crestani. Effects of word recognition errors in spoken query processing. In *Proc. IEEE Advances in Digital Libraries*, pages 39–47, Washington, DC, May 2000.

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

B. Decadt, J. Duchateau, W. Daelemans, and P. Wambacq. Transcription of out-of-vocabulary words in large vocabulary speech recognition based on phoneme-to-grapheme conversion. In *Proc. ICASSP '02*, Orlando, FL, May 2002.

S. Deligne and F. Bimbot. Inference of variable-length acoustic units for continuous speech recognition. *Speech Communication*, 23:"223–241", 1997.

S. Deligne, F. Yvon, and F. Bimbot. Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc. European Conf. on Speech Communication and Technology*, pages 2243–2246, Madrid, Spain, September 1995.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience, 2000.

N. Duta, R. Schwartz, and J. Makhoul. Analysis of the errors produced by the 2004 BBN speech recognition system in the DARPA EARS evaluations. *IEEE Trans. Acoustics, Speech and Language Processing*, 14(5):1745–1753, 2006.

A. C. Fang and M. Huckvale. Out-of-vocabulary rate reduction through dispersion-based lexicon acquisition. *Literary and Linguistic Computing*, 15(3):251–264, 2000.

E. Filisko and S. Seneff. Developing city name acquisition strategies in spoken dialogue systems via user simulation. In *Proc. SIGDIAL*, pages 144–155, Lisbon, Portugal, 2005.

E. Fosler, M. Weintraub, S. Wegmann, Y. H. Kao, S. Khudanpur, C. Galles, and M. Saraclar. Automatic learning of word pronunciation from data. In *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, October 1996.

O. Fujimura. Syllable as a unit of speech recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 23(1):82–87, 1975.

S. Furui, M. Nakamura, T. Ichiba, and K. Iwano. Why is the recognition of spontaneous speech so hard? In *Proc. 8th Intl. Conf. on Text, Speech, and Dialogue*, pages 9–22, Karlovy Vary, Czech Republic, September 2005.

L. Galescu. Recognition of out-of-vocabulary words with sub-lexical language models. In *Proc. European Conf. on Speech Communication and Technology*, pages 249–252, Geneva, Switzerland, September 2003.

L. Galescu and J. Allen. Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *Proc of the 4th ISCA Tutorial and Workshop on Speech Synthesis*, Pitlochry, Scotland, September 2001.

L. Galescu and J. Allen. Name pronunciation with a joint n-gram model for bi-directional grapheme-to-phoneme conversion. In *Proc. of ICSLP*, pages 109–112, Denver, Colorado, 2002.

A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. R. Doddington. Syllable-Based Large Vocabulary Continuous Speech Recognition. *IEEE Trans. Speech and Audio Processing*, 9(4):358–366, 2001.

J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren. NIST Rich Transcription 2002 Evaluation: A Preview. In *Proc. Intl Conf. on Language Resources and Evaluation*, pages 655–659, Canary Islands, Spain, 2002.

J. R. Glass. *Finding acoustic regularities in speech: applications to phonetic recognition.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1988.

J. R. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, pages 137–152, 2003.

J. R. Glass and V. Zue. Multi-level acoustic segmentation of continuous speech. In *Proc. ICASSP '88*, pages 429–432, New York, NY, April 1988.

J. R. Glass, J. Chang, and M. McCandless. A probabilistic framework for feature-based speech recognition. In *Proc. ICSLP '96*, volume 4, pages 2277–2280, Philadelphia, PA, October 1996.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP '92*, pages 517–520, San Francisco, CA, March 1992.

C. Gonzalez-Ferreras and V. Cardeoso-Payo. A system for speech driven information retrieval. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 624–628, Kyoto, Japan, December 2007.

Google. Web 1T 5-gram Version 1. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`.

A. Gorin, G. Riccardi, and J. Wright. How May I Help You? *Speech Communication*, 23:113–127, 1997.

O. Gospodnetic and E. Hatcher. *Lucene in Action.* Manning Publications, 2004.

D. Graff and M. Liberman. The 1996 broadcast news speech and language-model corpus. In *Proc. DARPA Speech Recognition Workshop*, pages 11–14, Chantilly, VA, February 1997.

S. Greenbaum. *Comparing English Worldwide: The International Corpus of English.* Oxford University Press, 1996.

S. Greenbaum and J. Svartvik. *The London-Lund Corpus of Spoken English*, chapter The London-Lund Corpus of Spoken English - Description and Research. Lund University Press, 1990.

S. Greenberg and B. E. D. Kingsbury. The modulation spectrogram: in pursuit of an invariant representation of speech. In *Proc. ICASSP '97*, pages 1647–1650, Munich Germany, April 1997.

A. H. Gruenstein and S. Seneff. Context-sensitive language modeling for large sets of proper nouns in multimodal dialogue systems. In *Proc. IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, 2006.

A. H. Gruenstein, B. J. Hsu, J. R. Glass, S. Seneff, L. Hetherington, S. Cyphers, I. Badr, C. Wang, and S. Liu. A multimodal home entertainment interface via a mobile device. In *Proc. Mobile Natural Language Processing*, pages 1–9, Columbus, Ohio, June 2008.

W. I. Hallahan. DECtalk software: text-to-speech technology and implementation. *Digital Technical Journal*, 7:5–19, 1995.

Y. Han, A. Hamalainen, and L. Boves. Trajectory clustering of syllable-length acoustic models for continuous speech recognition. In *Proc. ICASSP '06*, pages 1169–1172, Toulouse, France, May 2006.

A. Hausenstein. Using syllables in a hybrid HMM-ANN recognition system. In *Proc. European Conf. on Speech Communication and Technology*, pages 1203–1206, Rhodes, Greece, September 1997.

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

T. J. Hazen and I. Bazzi. A comparison and combination of methods for OOV word detection and word confidence scoring. In *Proc. ICASSP '01*, pages 397–400, Salt Lake City, UT, May 2001.

T. J. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Integrating recognition confidence scoring with language understanding and dialogue modeling. In *Proc. ICSLP '00*, pages 397–400, Beijing, China, October 2000a.

T. J. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Recognition confidence scoring for use in speech understanding systems. In *"Proc. ISCA Tutorial and Reseach Workshop"*, pages 49–67, Paris, France, September 2000b.

I. L. Hetherington. *"A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 1995.

L. Hetherington. An efficient implementation of phonological rules using finite-state transducers. In *Proc. European Conf. on Speech Communication and Technology*, pages 1599–1602, Aalborg, Denmark, September 2001.

L. Hetherington. The MIT finite-state transducer toolkit for speech and language processing. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 2609–2612, Jeju, South Korea, 2004.

K. Hofland and S. Johansson. How May I Help You? *The Norwegian Computing Center for the Humanities*, 1982.

Z. Hu, J. Schalkwyk, E. Barnard, and R. Cole. Speech recognition using syllable-like units. In *Proc. ICSLP '96*, pages 1117–1120, Philadelphia, PA, October 1996.

H. Van Der Hulst and N. Smith, editors. *The Structure of Phonological Representations - Part I*, chapter From cyclic phonology to lexical phonology. Foris Publications, Dordrecht, 1082.

H. Van Der Hulst and N. Smith, editors. *The Structure of Phonological Representations - Part II*, chapter The Syllable. Foris Publications, Dordrecht, 1982.

S. Y. Ishikawa, T. Ikeda, K. Miki, F. Adachi, R. Isotani, K. I. Iso, and A. Okumura. Speech-activated text retrieval system for multimodal cellular phones. In *Proc. ICASSP '04*, pages 453–456, Montreal, Canada, March 2004.

F. Jelinek, R. Mercer, and S. Roukous. Classifying words for improved statistical language models. In *Proc. ICASSP '90*, pages 621–624, Albuquerque, NM, April 1990.

L. Jiang, H. W. Hon, and X. Huang. Improvements on a trainable letter-to-sound converter. In *Proc. European Conf. on Speech Communication and Technology*, pages 605–608, Rhodes, Greece, September 1997.

R. J. Jones, S. Downey, and J. S. Mason. Continuous speech recognition using syllables. In *Proc. European Conf. on Speech Communication and Technology*, pages 1171–1174, Rhodes, Greece, September 1997.

D. Kahn. *"Syllable-based generalizations in English phonology"*. PhD thesis, Department of Linguistics and Philosophy, Cambridge, MA, September 1976.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

D. Klakow, G. Rose, and X. Aubert. OOV-detection in large vocabulary system using automatically defined word-fragments as fillers. In *Proc. European Conf. on Speech Communication and Technology*, pages 49–52, Budapest, Hungary, September 1999.

J. Kneissler and D. Klakow. Speech recognition for huge vocabularies by using optimized sub-word units. In *Proc. European Conf. on Speech Communication and Technology*, pages 69–72, Aalborg, Denmark, September 2001.

H. Kucera and W. N. Francis. *Computational Analysis of Present Day American English*. Brown University Press, 1967.

P. Ladefoged. *A course in phonetics*. Harcourt Brace Jovanovich Inc., 1975.

P. Lamere, P. Kwok, W. Walker, E. Gouveau, R. Singh, B. Raj, and P. Wolf. Design of the CMU SPHINX-4 decoder. In *Proc. European Conf. on Speech Communication and Technology*, pages 1181–1184, September 2003.

K. F. Lee, H. W. Hon, and R. Reddy. An overview of the SPHINX speech recognition system. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-38(1):35–45, 1990.

S. Lee and J. R. Glass. Real-time probabilistic segmentation for segment-based speech recognition. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 1803–1806, Sydney, Australia, December 1998.

H. Lin, J. Bilmes, D. Veryri, and K. Kirchhoff. OOV detection by joint word/phone lattice alignment. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 478–482, Kyoto, Japan, December 2007.

R. P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–15, July 1997.

K. Livescu and J. R. Glass. Segment-based recognition on the Phonebook task: initial results and observations on duration modeling. In *Proc. European Conf. on Speech Communication and Technology*, pages 1437–1440, Aalborg, Denmark, September 2001.

A. Ljolje, M. Saraclar, M. Bacchiani, M. Collins, and B. Roark. The ATT RT02 speech-to-text system. In *Proc. RT02 Workshop*, Vienna, VA, May 2002.

J. M. Lucassen and R. L. Mercer. An information theoretic approach to the automatic determination of phonemic baseforms. In *Proc. ICASSP '84*, pages 42.5.1–42.5.4, San Diego, California, March 1984.

K. Maekawa. Corpus of spontaneous Japanese: Its design and evaluation. In *"Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition"*, pages 7–12, Tokyo, Japan, 2003.

K. Maekawa, H. Kikuchi, and W. Tsukahara. Corpus of spontaneous Japanese: design, annotation, and XML representation. In *"Proc. Intl. Symp. on Large-scale knowledge resources"*, pages 19–24, Tokyo, Japan, 2004.

B. Maison. Automatic baseform generation from acoustic data. In *Proc. European Conf. on Speech Communication and Technology*, pages 2545–2548, Geneva, Switzerland, September 2003.

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

S. Mann, A. Berton, and U. Ehrlich. How to access audio files of large data bases using in-car speech dialogue systems. In *Proc. Interspeech*, pages 138–141, Antwerp, Belgium, August 2007.

C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

Y. Marchand and R. I. Damper. Multi-strategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26:195–219, 2000.

H. Meng. *"The use of distinctive features for automatic speech recognition"*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1995.

H. Meng, S. Seneff, and V. Zue. Phonological parsing for reversible letter-to-sound/sound-to-letter generation. In *Proc. ICASSP '94*, pages II1–II4, Adelaide, Australia, April 1994a.

H. Meng, S. Seneff, and V. Zue. Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation. In *Proc. of the Workshop on Human Language Technology*, pages 289–294, Plainsboro, NJ, March 1994b.

M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:269–312, 1997.

A. Moreno-Daniel, S. Parthasarathy, B. H. Juang, and J. G. Wilpon. Spoken query processing for information retrieval. In *Proc. ICASSP '07*, pages 121–124, Honolulu, Hawaii, April 2007.

Y. Muthusamy, R. Agarwal, Y. Gong, and V. Viswanathan. Speech-enabled information retrieval in the automobile environment. In *Proc. ICASSP '99*, volume 2, pages 2259–2262, Phoenix, AZ, March 1999.

P. Natarajan, R. Prasad, R. M. Schwartz, and J. Makhoul. A scalable architecture for directory assistance automation. In *Proc. ICASSP '02*, pages 21–24, Orlando, FL, May 2002.

L. Nguyen, B. Xiang, M. Afify, S. Abdou, S. Matsoukas, and R. Schwartz. The BBN RT04 English broadcast news transcription system. In *Proc. Interspeech*, pages 1673–1676, Lisbon, Portugal, September 2005.

N. J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann, 1980.

S. Oger, G. Linares, F. Bechet, and P. Nocera. On-demand new word learning using world wide web. In *Proc. ICASSP '08*, pages 4305–4308, Las Vegas, NV, April 2008.

M. Ostendorf. Moving beyond the beads on a string model of speech. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 79–83, Keystone, CO, December 1999.

A. Park and J. R. Glass. Unsupervised word acquisition from speech using pattern discovery. In *Proc. ICASSP '06*, pages 409–412, Toulouse, France, May 2006.

A. Park and J. R. Glass. Unsupervised pattern discovery in speech. *IEEE Trans. Acoustics, Speech and Language Processing*, 16(1):186–197, 2008.

D. B. Paul and J. M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proc. DARPA Speech and Natural Language Workshop*, pages 357–362, Harriman, NY, February 1992.

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

J. Pitrelli, C. Fong, S. Wong, J. Spitz, and H. Leung. Phonebook: A phonetically-rich isolated-word telephone-speech database. In *Proc. ICASSP '95*, pages 101–104, Detroit, MI, May 1995.

R. Prasad, S. Matsoukas, C. L. Kao, J. Z. Ma, D. X. Xu, T. Colthurst, O. Kimball, R. Schwartz, J. L. Gauvain, L. Lamel, H. Schwenk, G. Adda, and F. Lefevre. The 2004 BBN/LIMSI 20xRT English Conversational Telephone Speech Recognition System. In *Proc. Interspeech*, pages 1641–1644, Lisbon, Portugal, 2005.

P. Price. Evaluation of the spoken language systems: the ATIS domain. In *"Proc. DARPA Speech and Natural Language Workshop"*, pages 91–95, Hidden Value, CA, June 1990.

P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett. The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 651–654, New York, New York, April 1988.

Pronlex. CALLHOME American English Lexicon(PRONLEX). `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97L20`.

L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, 1993.

L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

M. A. Randolph. *"Syllable-based constraints on properties of English sounds"*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 1989.

R. Rosenfeld. Optimizing Lexical and N-gram Coverage via Judicious Use of Linguistic Data. In *Proc. European Conf. on Speech Communication and Technology*, pages 1763–1766, Madrid, Spain, September 1995.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

T. Schaaf. Detection of OOV words using generalized word models and a semantic class language model. In *Proc. European Conf. on Speech Communication and Technology*, pages 2581–2584, Aalborg, Denmark, September 2001.

O. Scharenborg and S. Seneff. A two-pass strategy for handling OOVs in a large vocabulary recognition task. In *Proc. Interspeech*, pages 1669–1672, Lisbon, Portugal, September 2005.

H. Schramm, B. Rueber, and A. Kellner. Strategies for name recognition in automatic directory assistance systems. *Speech Communication*, 31:329–338, 2000.

E. G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck. Acoustic modelling of subword units in the ISADORA speech recognizer. In *Proc. ICASSP '92*, pages 577–580, San Francisco, CA, March 1992.

S. Seneff. Response planning and generation in the MERCURY flight reservation system. *Computer Speech and Language*, 16:283–312, 2002.

S. Seneff. Reversible sound-to-letter/letter-to-sound modeling based on syllable structure. In *Proc. NAACL-HLT*, Rochester, NY, April 2007.

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

S. Seneff. ANGIE: A new framework for speech analysis based on morph-phonological modeling. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 110–113, Philadelphia, PA, October 1996.

S. Seneff, H. Meng, and V. Zue. Language modeling for recognition and understanding using layered bigrams. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 317–320, Alberta, Canada, October 1992.

S. Seneff, R. Lau, and H. Meng. ANGIE: A new framework for speech analysis based on morpho-phonological modelling. In *Proc. ICSLP '96*, volume 1, pages 110–113, Philadelphia, PA, 1996.

A. Sethy, B. Ramabhadran, and S. Narayanan. Improvements in English ASR for the MALACH project using syllable-centric models. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 129–134, St. Thomas, Virgin Islands, December 2003.

T. Sloboda and A. Waibel. Dictionary learning for spontaneous speech recognition. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 2328–2331, Philadelphia, PA, October 1996.

F. K. Soong and E. Huang. A tree-trellis based fast search for finding the N-best sentence hypotheses in continuous speech recognition. In *Proc. ICASSP '91*, pages 705–708, Toronto, Canada, May 1991.

B. Suhm, M. Woszczyna, and A. Waibel. Detection and transcription of new words. In *Proc. European Conf. on Speech Communication and Technology*, pages 2179–2182, Berlin, Germany, September 1993.

M. Thomae, T. Fabian, R. Lieb, and G. Ruske. Lexical out-of-vocabulary models for one-stage speech interpretation. In *Proc. Interspeech*, pages 441–444, Lisbon, Portugal, September 2005.

Vlingo. Revolutionizing Voice UI for Mobile, May 2008. Vlingo Unconstrained Speech Recognition White Paper.

Y. Y. Wang, D. Yu, Y. C. Ju, and A. Acero. An introduction to voice search. *IEEE Signal Processing Magazine*, 25(3):28–38, May 2008.

C. Wayne. Effective, affordable, reusable speech-to-text, May 2003. presented at the EARS 2003 Meeting.

R. Weide. The CMU pronunciation dictionary, 1998. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

C. M. Westendorf and J. Jelitto. Learning pronunciation dictionary from speech data. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 1045–1048, Philadelphia, PA, October 1996.

E. Whittaker and P. Woodland. Particle-based language modelling. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 170–173, Beijing, China, October 2000.

P. Wolf and B. Raj. The MERL SpokenQuery information retrieval system: a system for retrieving pertinent documents from a spoken query. In *Proc. on Multimedia and Expo*, pages 317–320, Lausanne, Switzerland, August 2002.

P. Wolf, J. Woelfel, J. Van Gemert, B. Raj, and D. Wong. SpokenQuery: An alternate approach to choosing items with speech. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 221–224, Jeju Island, Korea, October 2004.

S. Wu, B. E. D. Kingsbury, N. Morgan, and S. Greenberg. Incorporating information from syllable-length time scales into automatic speech recognition. In *Proc. ICASSP '98*, pages 721–724, Seattle, WA, May 1998a.

S. Wu, B. E. D. Kingsbury, N. Morgan, and S. Greenberg. Performance improvements through combining phone- and syllable-scale information in automatic speech recognition. In *Proc. ICSLP '98*, pages 459–462, Sydney, Australia, December 1998b.

A. Yazgan and M. Saraclar. Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition. In *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 745–748, Montreal, Canada, 2004.

S. R. Young. Detecting misrecognitions and out-of-vocabulary words. In *Proc. ICASSP '94*, pages 21–24, Adelaide, Australia, April 1994a.

S. R. Young. Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words. Technical Report 157, Carnegie Mellon University, Pittsburg, PA, May 1994b.

L. Zhang. Speech recognition using syllable and pseudo articulatory features modeling. In *Proc. Natural Language Processing and Knowledge Engineering*, pages 137–141, Wuhan, China, October 2005.

V. Zue, J. R. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. The VOYAGER speech understanding system: a progress report. In *Proceedings of the workshop on Speech and Natural Language*, pages 51–59, Cape Cod, MA, oct 1989a.

V. Zue, J. R. Glass, M. Phillips, and S. Seneff. The MIT SUMMIT speech recognition system: a progress report. In *Proc. Speech and Natural Language Workshop*, pages 179–189, Philadelphia, PA, February 1989b.

V. Zue, J. R. Glass, D. Goodine, M. Phillips, and S. Seneff. The SUMMIT speech recognition system: phonological modelling and lexical access. In *Proc. ICASSP '90*, pages 49–52, Albuquerque, NM, April 1990.

V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington. JUPITER: A telephone-based conversational interface for weather information. *IEEE Trans. Speech and Audio Processing*, 8(1):85–96, January 2000.