1)

A Characterization of the Problem of New, Out-of-Vocabulary Words in Continuous-Speech Recognition and Understanding

by

Irvine Lee Hetherington

S.B. and S.M., Massachusetts Institute of Technology (1989)

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

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1995

Author ....
Department of Electrical Engineering and Computer Science
October 13, 1994

Certified by ....
Victor W. Zue
Senior Research Scientist
Thesis Supervisor

Accepted by ....
F. R. Morgenthaler
Chair, Departmental Committee on Graduate Students

A Characterization of the Problem of New, Out-of-Vocabulary Words in Continuous-Speech Recognition and Understanding

by

Irvine Lee Hetherington

Submitted to the Department of Electrical Engineering and Computer Science
on October 13, 1994, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis is directed toward the characterization of the problem of new, out-of-vocabulary words for continuous-speech recognition and understanding. It is motivated by the belief that this problem is critical to the eventual deployment of the technology, and that a thorough understanding of the problem is necessary before solutions can be proposed. The first goal of this thesis is to demonstrate the magnitude of the problem. By examining a wide variety of speech and text corpora for multiple languages, we show that new words will always occur, even given very large system vocabularies, and that their frequency depends on the type of recognition task. We classify potential new words in terms of their phonological and syntactic characteristics. The second goal of this thesis is to characterize recognizer behavior when new words occur. We demonstrate that not only are new words themselves misrecognized, but their occurrence can cause misrecognition of in-vocabulary words in other parts of an utterance due to contextual effects. To perform our recognition study of new words, we developed an algorithm for efficiently computing word graphs. We show that word graph computation is relatively insensitive to the position of new words within an utterance. Further, we find that word graphs are an effective tool for speech recognition in general, irrespective of the new-word problem. The third and final goal of this thesis is to examine the issues related to learning new words. We examine the ability of the (context-independent) acoustic models, the pronunciation models, and the class n-gram language models of the SUMMIT system to incorporate, or learn, new words; we find the system's learning effective even without additional training. Overall, this thesis offers a broad characterization of the new-word problem, describing in detail the magnitude and dimensions of the problem that must be solved.

Thesis Supervisor: Victor W. Zue
Title: Senior Research Scientist

Acknowledgements

I wish to thank Victor Zue, my thesis supervisor, for the guidance and support he has provided me. He has offered a good balance between needed guidance and the latitude to do my own research. He has created an excellent group of researchers, students and staff, that make up the Spoken Language Systems Group. I am certain that his inviting me to join the group as a student five years ago has forever changed my career for the better. I look forward to continuing my work with Victor as I join the group as staff.

I would like thank my thesis committee, Victor Zue, Stephanie Seneff, and Eric Grimson, for their input on my thesis work. Their comments have helped me to improve this document in many ways. I thank them for their time and patience.

The members of the Spoken Language Systems Group deserve many, many thanks. Without daily assistance and discussion this thesis would not have been possible. Mike Phillips deserves special thanks for his countless hours of help with the recognizer and associated tools. He and Jim Glass helped me greatly with the development of the word graph representation and search algorithm. I very much enjoyed working with Hong Leung on the development of the Stochastic Explicit-Segment Model recognizer before this thesis research got under way. Nancy Daly deserves thanks for teaching me almost everything I know about spectrogram reading. I look forward to working more closely with Stephanie Seneff on the problems associated with new words and their interaction with natural language understanding. My past officemates, Jeff Marcus and Partha Niyogi, took part in hours of technical discussions, some of them quite lively. Jeff, due to his insight about statistical modeling and methods, has improved my own understanding. Giovanni Flammia, as well as being an entertaining officemate, collected and provided me with the Italian Voyager corpus. Mike McCandless helped to compute word graph statistics and has helped me to understand some of the subtle details of the recognition system. Eric Brill helped me to tag utterances using the part-of-speech tagger he developed. Dave Goddeau has helped me to understand and use his n-gram language models. Joe Polifroni, Christine Pao, and David Goodine have always been there when I have had Sun machine/system problems and when I needed to find

just one more file or CD-ROM. Rob Kassel has always answered my Macintosh-related questions. Sally Lee really came through in a pinch, getting a draft of my thesis to two thirds of my committee halfway around the world even though they were a moving target. Vicky Palay has administered the group so efficiently that it seems to run itself, but she deserves most of the credit. I have been extremely fortunate to be a member of the Spoken Language Systems Group and am even more fortunate to be able to remain with the group after graduation.

Thanks to my parents for enabling me to go to MIT in the first place as an undergraduate. They have always encouraged me and filled me with confidence. I have enjoyed spending time with my brother Kevin during his years at MIT.

My wife Sarah deserves the biggest thanks of all. She has supported me emotionally through good times and bad over the past nine years, almost my entire stay at MIT. I cannot really put into words what she has meant to me, nor can I imagine my life without her. Finally, thanks to our new son Alexander for providing the final impetus to finish this thesis. Of course, he came along before I finally finished. (Don't they always?)

This research was supported by ARPA under contract N00014-89-J-1332, monitored though the Office of Naval Research, by a research contract from NYNEX Science and Technology, and by a grant from Apple Computer, Inc.

To Sarah and Alexander

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Although current spoken language systems show great promise toward providing useful human-machine interfaces, they must improve substantially in terms of both accuracy and robustness. Lack of robustness is perhaps the biggest problem of current systems. To be robust, a system must be able to deal with, among other things, spontaneously produced speech from different speakers. Such spontaneous speech typically contains hesitations, filled pauses, restarts, and corrections, as well as well-formed words that are outside of the system's vocabulary. It is understanding the problem of these out-of-vocabulary words, or *new words,* that is the focus of this thesis research. This problem is one that must be thoroughly addressed before speech recognition systems can fully handle natural speech input in a wide variety of domains.

We believe that the new-word problem is much more important than is apparent from the relatively limited amount of research on the topic thus far. As we will see, it is virtually impossible to build a system vocabulary capable of covering all the words in input utterances. For any task other than one with a very small vocabulary, it is impractical to present users with a list of allowed words. Users will, in all likelihood, not be willing or able to memorize such a list and will invariably deviate from that list unknowingly. If a speech recognition system is not designed to cope with new words, it may simply attempt to match the acoustics of a new word using combinations of in-vocabulary words; the recognized string of words will contain errors and may not make sense (e.g., substituting "youth in Asia" for "euthanasia"). In an interactive problem-solving environment, the system will either perform some unintended action or reject the utterance because it is unable to understand the sentence. In both situations, the user will likely not know which word is at fault and may continue to use the word, causing further system confusion and user frustration.

Detecting and localizing new words in the input utterance could greatly improve system responses by allowing valuable feedback to the user (e.g., "I heard you ask for the address of a restaurant that I don't know. Can you spell it for me?"). If the system can maneuver the user back into the allowable vocabulary quickly, the interactive session could be more productive and enjoyable.

After detecting new words, adding them to the system would allow them to be treated as normal in-vocabulary words. Without such learning, the vocabulary must be tailored to minimize new words during system use and testing. A system capable of learning new words would make initial vocabulary determination less critical since its vocabulary would be adaptive. Such a system may be able to use a smaller vocabulary since it could rely on detection and learning to handle the increased number of new words. This is the goal in solving the new-word problem.

1.1 Background

Automatic speech recognition is the task of decoding a sequence of words from an input speech signal. In some systems, not only is the speech transcribed, such as in a dictation system, but it is understood by the system using some domain-specific knowledge. Automatic speech recognition and understanding can be extremely useful, since speech is a very efficient and natural mode of communication. Ideally, a person could walk up to a system and, in natural language, request information or instruct the computer to perform a desired task. Not only could speech be a convenient computer user interface, but in the case of telephone communication or times when the hands are not free, it is almost a necessity.

1.1.1 Speech Recognition Basics

Speech recognition is typically formulated as a maximum a posteriori (MAP) search for the most likely sequence of words, given appropriate acoustic measurements, where the words are drawn from the system's vocabulary. The sequence of words that has the highest a posteriori probability based on available acoustic measurements and linguistic constraints is chosen as the recognizer's top-choice hypothesis. Pruning during the search is critical since the search space of possible word sequences is so large: $O(\ell^v)$, where v is the vocabulary size and ℓ is the sequence length.

Figure 1-1 shows a block diagram of a generic speech recognition/understanding system. A signal processing component computes a set of acoustic measurements for each segment of speech. In the case of frame-based systems (e.g., hidden Markov models or HMMs [61]), the segments are simply fixed-rate frames. In the case of a segmental system, these segments are typically of variable duration and may overlap one another. The *acoustic models* generally model sub-word units such as phones and may be context-independent or context-dependent. The *lexical models* model the pronunciation of words in the vocabulary and constrain the sequence of sub-word units. The *language model,* often a statistical n-gram [33], constrains the word order. The search component makes use of acoustic, lexical, and language models to score word sequences. Typically, the best N complete-sentence hypotheses are determined. These best hypotheses are often fed into a natural language system, where they may be filtered for syntactic/semantic worthiness and/or a meaning representation may be generated. In some systems, the

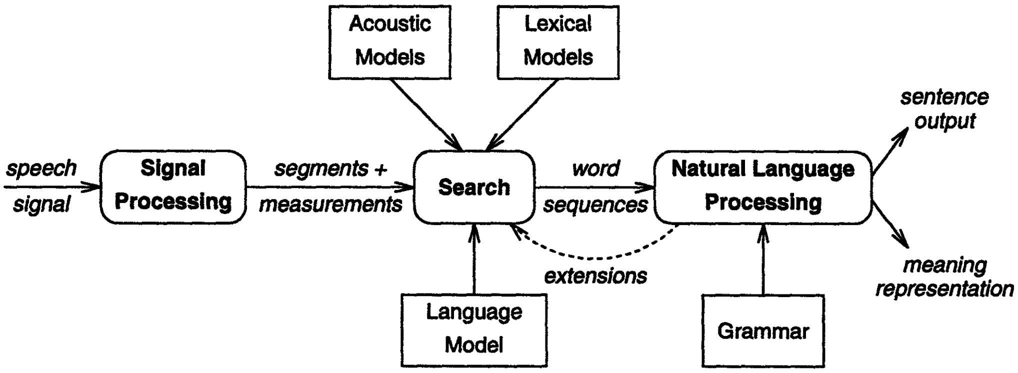


Figure 1-1: Block diagram of generic speech recognition/understanding system.

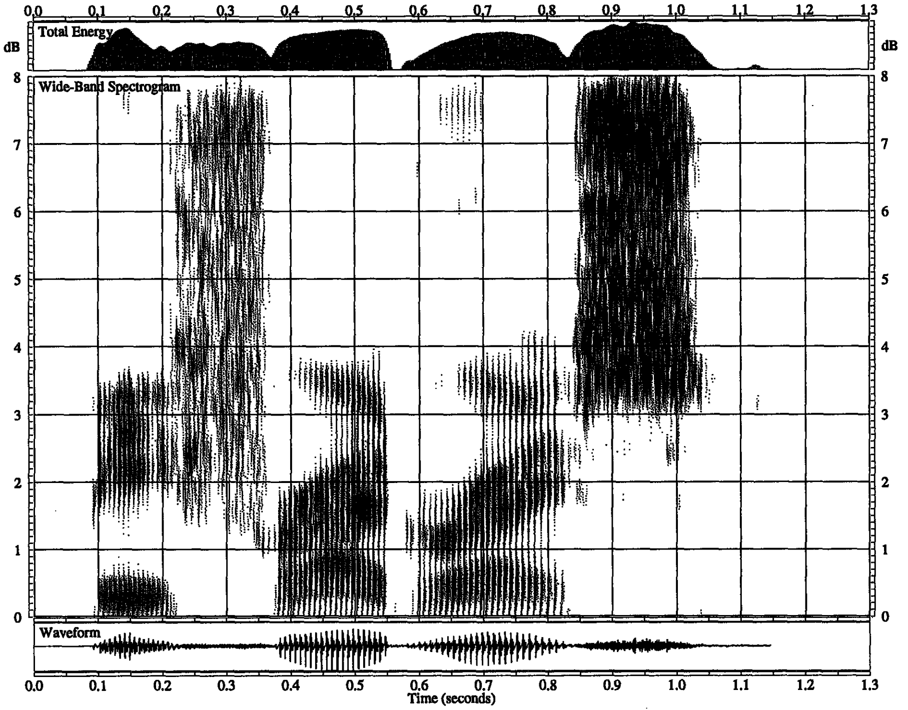


Figure 1-2: Example of continuous speech. The utterance is "beef fried rice" and shows that word boundaries are not readily apparent in continuous speech. The word "beef" spans 0.09–0.28 s, "fried" spans 0.28–0.60 s, and "rice" spans 0.60–1.05 s. The words "beef fried" are joined by a geminate /f/, which is longer than a normal /f/. The range 0.55–0.58 s, which appears to be a pause, is the closure for the /d/ in "fried."

natural language system is closely integrated into the search [26], providing linguistically sensible word extensions to restrict the search space.

The ideal speech recognition system is speaker-independent, has a large vocabulary, and can operate on spontaneous, continuous speech. Historically, systems have been simplified along several of these dimensions in order to achieve acceptable performance, in terms of both accuracy and speed. Speaker-dependent systems must be trained on speech from the speaker(s) who will be the eventual users of the system. The result may be increased accuracy on those speakers at the expense of a required speaker enrollment period and a less flexible system. A smaller vocabulary reduces the amount of computation, since there are fewer word sequences to be considered, and hopefully increases accuracy on the in-vocabulary words due to there being fewer confusable words. The primary cost of a smaller vocabulary is an increased number of new, out-of-vocabulary words. This is an issue we will examine in this thesis. Recognizing isolated-word speech is significantly easier than recognizing continuous speech. In isolated-word speech the speaker pauses briefly between words. In continuous speech there are not generally pauses between words, making even the task of finding word boundaries difficult, as can be seen in Figure 1-2. Finally, spontaneous speech is filled with effects that are not present in read speech, in which someone is reading from a script. There are hesitations, filled pauses (e.g., "um," "uh," etc.), and false starts (e.g., "I want to fly to Chi— yeah, to Chicago").

1.1.2 State of the Art

Current state-of-the-art systems are speaker-independent, large-vocabulary, continuous-speech systems. For example, in December of 1993, fourteen sites, four of them outside the U. S., took part in Advanced Research Project Agency's (ARPA) evaluation of speech recognition and understanding systems. The Air Travel Information Service (ATIS) task [17, 59] was used for recognition and understanding and consisted of spontaneous utterances regarding airline flight information. The Wall Street Journal (WSJ) task [55] was used for recognition only and consisted of read speech drawn from newspaper articles from the *Wall Street Journal.* The vocabulary size used for ATIS was on the order of 2,500 words, and the size used for WSJ ranged from 5,000 to 40,000 words.

The best speech recognition performance on the ATIS task is 3.3% word-error rate[1] on "answerable" utterances.[2] This means that, on the average, these systems are making fewer than one error every twenty-five words. The error rate for complete sentences is now about 18% on answerable utterances, meaning that only about one in five sentences will contain a recognition error. Three years ago, the sentence error rate was nearly three times larger on the same task but with a smaller vocabulary. On the WSJ task, the lowest word-error rate for a 20,000-word system was 11.2%, and for a 5,000-word system the lowest was 5.3% [23].

[1] The word-error rate takes into account, for each utterance, the number of word substitutions, deletions, and insertions. The %word-error is defined as %substitutions + %deletions + %insertions.

[2] The "answerable" utterances included the class "A" (dialog-independent) and class "D" (dialog-dependent) utterances. The class "X" utterances, which are essentially out-of-domain, were excluded.

1.1.3 New-Word Problem Artificially Reduced

Although the ARPA program has greatly promoted speech recognition and understanding research through the definition of "common tasks" and data collection for them, it is our belief that most of these common tasks have downplayed the importance of the new-word problem. The first ARPA common task for speech recognition was the Resource Management (RM) task [58]. The speech data consisted of read speech from a naval resource management domain, in which the scripts used during data collection were *generated* by an artificial language model. This language model, a finite-state grammar, generated sentences with a closed, or limited, vocabulary. Thus the RM task completely side-stepped the new-word problem. This is understandable, since this was an early attempt at a common task to further speech recognition technology. However, enforcing a closed vocabulary hides a problem we expect a real system to face.

The recent ARPA WSJ evaluation was divided into two conditions: 5,000- and 20,000-word vocabulary sizes. In both cases, the frequency of new words was artificially reduced. For the 5,000-word small-vocabulary condition, the vocabulary was completely closed, meaning that there were *zero* new words. For the 20,000-word large-vocabulary condition, all training and testing data were artificially *filtered* to contain only words from the set of 64,000 most-frequent words. Since the task vocabulary was larger than the system vocabulary in this condition, the systems did face some new words. However, the vocabulary filtering artificially reduced their frequency. For example, with the 20,000-word vocabulary 2.27% of the words in the development set were out-of-vocabulary [71]. If the vocabulary was increased to contain the 40,000 most-frequent words, the percentage of new words fell to only 0.17%. As we shall see in Chapter 2 of this thesis, these new-word rates, particularly that corresponding to the 40,000-word vocabulary, were artificially low due to the 64,000-word filtering.[3] Of course, the vocabulary filtering was possible because WSJ utterances are read speech collected with prescribed scripts. With more realistic spontaneous speech such filtering would be im-

[3] In Chapter 2, we estimate the new-word rates to be 2.4% and 1.1% for 20,000- and 40,000-word vocabularies, respectively.

possible and the new-word rate would certainly be higher.[4]

ATIS is more realistic in that it was collected spontaneously and the utterances were not filtered based on vocabulary. However, the limited scope of the task, in terms of both the number of cities included (46) and the manner in which the data were collected may keep the frequency of new words low. Most of the ATIS data were collected with users trying to solve prescribed travel-planning problems or scenarios. Many of the scenarios mentioned specific cities, effectively steering users toward the allowed set of cities. We would expect that the scenarios greatly reduce the number of new words from users asking about cities and airports outside of the official ATIS domain. However, the fact that ATIS is collected spontaneously from users, and that it is not filtered based on vocabulary means that ATIS is a step in the right direction toward more realism.

1.2 Prior Research

When this thesis was initiated in 1992, very little research on the new-word problem had been reported. Since that time, more has begun to surface, suggesting that researchers are beginning to realize that the new-word problem is one that must be addressed. In this section, we discuss reported research on the new-word problem and closely related fields. We divide the new-word research into three areas: characterization of the problem, the detection problem, and the learning problem. The detection problem involves recognizing that an utterance contains out-of-vocabulary word(s) and locating these words. The learning problem involves incorporating new words into a system so they become a part of the system's working vocabulary.

1.2.1 Characterization of the Problem

Characterizing the new-word problem in a general manner is a logical first step. Unfortunately, the literature is lacking in this subject. It seems that many researchers in the field attempt to solve the problem without first demonstrating the magnitude of

[4] In fact, in a subset of WSJ containing spontaneously produced dictation, 1.4–1.9% of the words were out-of-vocabulary for a 40,000-word vocabulary [46].

the problem, characterizing new words and their usage, and quantifying their effects on recognition (without detection).

However, the work of Suhm et al. [69] is an exception.[5] They chose to characterize the problem before attempting to solve the detection problem. Based on orthographic transcriptions in the Wall Street Journal (WSJ) domain they performed a study of the new-word problem by examining vocabulary growth, vocabulary coverage (i.e., new-word rate), characteristics of new words, and issues related to modeling new words within a statistical n-gram language model. (See Section 1.2.2 for a discussion of their work on the detection problem.)

In their study, Suhm et al. used a variable number of training sentences to automatically determine various vocabularies that resulted in 100% coverage on the training material. Over the range of training data sizes they examined, from 250 to 9000 sentences, the vocabulary grew from 1,721 to 14,072 words while the new-word rate on an unseen test set fell from 27.2% to 4.2%. This result demonstrates that even fairly large vocabularies can still result in a significant new-word rate on unseen data.

Suhm et al. classified more than 1,000 new words that had not been covered by their largest vocabulary and found that 27% were names, 45% were inflections of in-vocabulary words, and 6% were concatenations of in-vocabulary words. This means that about half of the new words could be built from in-vocabulary words. This significant result implies that a system capable of automatically handling inflections and concatenations may be able to handle a large fraction of new words. Perhaps system vocabularies should be more than merely a list of distinct words.

In further characterizing new words, Suhm et al. examined their length as measured by number of phonemes. They found that the length of new words was significantly longer than the overall (frequency-weighted) length of all words. However, when compared to the vocabulary in an unweighted fashion, the length distribution was very similar.

Suhm et al. also studied the introduction of a new-word class into a statistical word

[5] The study of Suhm et al. [69] was reported concurrently with our initial study [28] at Eurospeech '93. However, their study is less general in that it involved only one language (English) and one domain (Wall Street Journal).

trigram language model. In this study they mapped all out-of-vocabulary words to the new-word class. In order to evaluate the language model's ability to constrain the location of new words and to model the words that occur after new words, they introduced a few perplexity-like measures.[6] These measures were an attempt to quantify detection and false-alarm characteristics at the language model level (i.e., text only) in terms of language model constraint as measured by perplexity. However, since the resulting values are so unlike the overall WSJ trigram perplexity they report, and since no one else has used them to our knowledge, they are difficult to interpret.

1.2.2 Detecting New Words

Asadi et al. reported some of the earliest research into the problem of detecting new words [2,4]. They also examined the learning problem (see Section 1.2.3). Their research was carried out on the Resource Management (RM) task [58], using the BBN BYBLOS continuous speech recognition system [4,14,15]. The BYBLOS system used HMMs and a statistical class bigram language model.

It is important to note that because the utterances in the RM task were generated artificially from a finite-state grammar, there were no true new words. All new words for their experiments were simulated by removing words from the open semantic classes, namely ship names, port names, water names, land names, track names, and capabilities. See Table 1-1 for the simulated new words and their classes. Of the 55 words removed from the normal RM vocabulary, 90% were proper nouns and their possessives.[7]

For detection, they experimented with different acoustic models for new words. These models were networks of all phone models with an enforced minimum number of phonemes (2–4). Asadi et al. tried both context-independent and context-dependent phonetic models. The statistical class bigram language model of the BYBLOS system allowed them to enable new words precisely where they were appropriate for the task: in open semantic classes. Since they simulated the new words by removing words from specific classes, they knew exactly where to allow them in their semantic class bigram.

[6] Perplexity L is related to entropy H by $H = \log L$, where $H = -\sum_x P(x) \log P(x)$, [33].

[7] As we will see in Chapter 2, this is not typical for true new words.

class	examples
ship name	Chattahoochee, Dale, Dubuque, England, Firebush, Manhattan, Sacramento, Schenectady, Vancouver, Vandergrift, Wabash, Wadsworth, Wasp
port name	Aberdeen, Alaska, Alexandria, Astoria, Bombay, Homer, Oakland, Victoria
water name	Atlantic, Bass, Bering, Coral, Indian, Korean, Mozambique, Pacific, Philippine
land name	California, French, Korea, Philippines, Thailand
track name	DDD992
capability	harpoon, NTDS, SPS-40, SPS-48, SQQ-23, TFCC

Table 1-1: New words simulated by Asadi et al. in the RM task. In addition, the possessive forms of the ship names were included as well.

It is likely that their language model provided more constraint on new words than would otherwise be expected with real, non-simulated new words.

Overall, Asadi et al. found that an acoustic model requiring a sequence of at least two context-independent phonemes yielded the best detection characteristics: between 60–70% detection rate with a 2–6% false-alarm rate. They found that the new-word model consisting of context-dependent phoneme models, although considerably more complex computationally, resulted in a higher false-alarm rate without a significant increase in the detection rate. They attributed this to the fact that the system used context-dependent phoneme models for in-vocabulary words, and thus the new-word model tended to trigger inappropriately during in-vocabulary speech. In effect, they found it advantageous to bias the system away from new words by using less-detailed acoustic models for them.

At the time this research was conducted, the RM task was a natural choice. It was a contemporary task, and there was a relatively large quantity of data available for experimentation. However, the artificial nature of the utterance scripts casts doubt on the realism of the new words studied. Nonetheless, this was pioneering research.

Kita et al. [38] experimented with new-word detection and transcription in continuous Japanese speech using HMMs and generalized LR parsing. Basically, they use two models running in parallel, one with a grammar describing their recognition task and the other with a stochastic grammar describing syllables in Japanese. They used context-independent phone models throughout. The output of the system was a string of in-vocabulary words and strings of phones in which out-of-vocabulary words occurred. Results were not very encouraging: when the new-word detection/transcription capability was enabled, overall word-error rate increased from 15.8% to 18.3%. The benefit of having a new-word model was overshadowed by false alarms in regions where in-vocabulary words occurred.

One potential problem with this work that prevents generalization from it was the relatively small amount of data used in the investigation. The task was an international conference secretarial service. Because all of their data was in-vocabulary, Kita et al. had to remove words in order to simulate new words. To evaluate their new-word detection and transcription technology, they removed only eight words, all proper nouns. In their evaluation data, they had only fourteen phrases that contained new words. The number of new-word occurrences was so small that it is difficult to draw any conclusions from their results.

Itou et al. [31] also performed joint recognition and new-word transcription in continuous Japanese speech. They used HMMs with context-independent phone models and a stochastic phone grammar. Overall, their system achieved a correct-detection rate of 75% at a false-alarm rate of 11%.

The task was not described at all, except that it contained 113 unique words and the system was speaker-independent and accepted continuous speech. They removed six words from the lexicon, each from a different category of their grammar. This resulted in 110 out of 220 test utterances containing new words. However, because neither the task not the selection of simulated new words is described adequately, it is difficult to interpret their reported level of detection performance.

Suhm et al. [69], in addition to providing one of the very few characterizations of the new-word problem, experimented with detection in English. In the context of a conference registration task, they examined detection and phonetic transcription of new words. This was not the same task they used in their initial study. Since the set of available training data was small, they used a word bigram with new-word capability instead of a trigram as they had used in their WSJ text study. The test set consisted of 59 utterances containing 42 names. All names were removed from the vocabulary to simulate new words, thus leaving them with 42 occurrences of new words.[8] For

[8] Given that in their WSJ study they reported that only 27% of new words were names, it is curious

detection, they achieved a 70–98% detection rate with a 2–7% false-alarm rate. For phonetic transcription of new words, they achieved a phonetic string accuracy of 37.4%.

1.2.3 Learning New Words

Learning new words involves updating the various components of a system so that the previously unknown words become a part of the system's working vocabulary. The acoustic, lexical, and language models all may need to be updated when a new word is to be added to a system. If adding words to a system were easy, perhaps even automatic, then the system vocabularies could better adapt to the tasks at hand.

Jelinek et al. [34] studied the problem of incorporating a new word into a statistical word n-gram language model. Such a model typically requires a very large amount of training data to estimate all the word n-tuple frequencies. The problem researchers encounter is that there is generally very little data available that includes the new word with which to update the n-gram frequencies effectively.

Jelinek et al.'s approach to solving the problem was to assign a new word to word classes based on its context. By comparing the context of the new word to the contexts of more frequent words, they were able to identify "statistical synonyms" of the new word. The probabilities for the synonyms were combined to compute the trigram probability model of the new word. They found that this synonym-based approach was vastly superior to a more straightforward approach based on a single new-word class in the language model. The synonym approach reduced perplexity measured on the new words by an order of magnitude while not increasing significantly the perplexity measured on the in-vocabulary words. This is a powerful technique for incorporating a new word into a statistical language model. This work represented an extension of the thesis work of Khazatsky at MIT [37].

Asadi et al. [3–5] studied the problem of building lexical (pronunciation) models for new words. They experimented with automatic phonetic transcription and a text-to-speech system (DECtalk) that generated pronunciations given new-word spellings.

that they chose to make *all* of their new words be names for their detection study. It would have been more realistic to study non-name new words too.

They found that with phonetic transcription, even using context-dependent triphone models and a phone-class bigram "language model," their results were inferior to those generated from a transcription using DECtalk's rule-based letter-to-sound system. As an interesting extension, they combined the two methods in order to further improve the pronunciation models. They generated a transcription using DECtalk, and then, using a probabilistic phone confusion matrix, expanded the transcription into a relatively large pronunciation network. This network was used to constrain the automatic phonetic transcription given a single utterance of the new word. They found this hybrid method produced transcriptions of comparable quality to those produced manually.

Once they generated an acoustic model of the new word, they added the new word to the statistical class grammar. This was easy for them because the task was defined in terms of a semantic class grammar. Thus, they simply added the new word to the appropriate class. The fact that they had little trouble adding new words to the language model is probably due to the fact that the RM task was generated using an artificial semantic finite-state grammar that was well-modeled by their statistical class bigram language model. In general, the problem of adding new words to a language model is probably more difficult, as evidenced by the work of Jelinek et al. However, it is evident that a language model defined in terms of word classes may be advantageous for learning.

Brill [11] studied the problem of assigning part-of-speech to new words. This work was a component of a part-of-speech tagger used to tag large bodies of text automatically. The system needed to assign tags to new words that were not seen in training material. Brill's part-of-speech tagger was based on automatically learned transformational rules and achieved a tagging accuracy, for twenty-two different tags, of 94% on all words and 77% on new words. Thus, there is hope for automatically deducing some syntactic properties of new words after detection. Clearly, the context of a new word can help in classifying its part-of-speech and perhaps other features.

Sound-to-letter and letter-to-sound systems are closely related to the issue of new-word learning. Automatic phonetic transcription followed by sound-to-letter conversion could be used to generate spellings of new words automatically. Conversely, if the user types in the spelling of a new word when adding it to a system, letter-to-sound rules could be used to generate a pronunciation model of it as Asadi et al. [3–5] did.

Meng et al. [30, 42, 43] developed a reversible letter-to-sound/sound-to-letter generation system using an approach that combined a multi-level rule-based formalism with data-driven techniques. Such a reversible system could be particularly useful in the context of learning new words, because both directions could be put to use. We should point out that by "sound," they meant phonemes plus stress markers.

Alleva and Lee [1] developed an HMM-based system in which they modeled the acoustics of letters directly. Associated with each context-dependent letter, a letter trigram, was an HMM model. Sound-to-letter was achieved by decoding the most likely sequence of letters directly, eliminating the need to go through the intermediate step of phonetic recognition. However, the phonetic transcription of a new word could be useful in building a pronunciation model.

A potential complication in the attempt to deduce the spelling of a detected new word is that the endpoints of the new word may be difficult to determine. Additionally, between words, pronunciation can be affected by the identity of the phones at the boundary. For example, "did you" is often pronounced as [dɨǰə] as opposed to the more canonical [dɨdyu^{w}]. Here, the realization of both words has been affected. Such phonological effects at the boundaries of new words will also complicate the precise location of them during detection.

1.2.4 Comments

There was virtually no work on the problem of new words before Asadi et al. [2] first investigated the detection problem. Since then, the amount of research on the detection and learning problems has increased. While this is encouraging, we feel that the new-word problem is still not getting the attention it deserves.

The work by Suhm et al. [69] includes a characterization of the new-word problem that is lacking in some of the other prior research. This work included a study of the frequency and length characteristics of new words for a subset of the WSJ corpus. While this research is a step in the right direction, Suhm et al. only examined a single corpus.

If we hope to characterize the general problem of new words, we need to examine multiple corpora from wide-ranging tasks. Furthermore, it is important to conduct carefully controlled experiments so as to separate the effects of new words from other recognition and understanding errors. We must thoroughly understand the new-word problem before we can hope to solve it in general.

1.3 Thesis Goals

The primary goal of this thesis is to *examine* the new-word problem to understand its magnitude and dimensions. This thesis is intended to fill some of the gaps in the prior research. We feel that a thorough understanding of the problem is required before trying to solve the detection and learning subproblems. The goals of this thesis are as follows:

1. *Demonstrate the magnitude of the new-word word problem.* As we have pointed out, we feel that the problem has not received the attention it deserves. We intend to demonstrate the seriousness of the new-word problem in a wide variety of tasks.

2. *Characterize new words in terms of their lexical, phonological, syntactic, and semantic characteristics.* We feel that it is important to understand the characteristics of new words so that they may ultimately be modeled effectively.

3. *Characterize recognizer behavior when faced with new words.* To understand the magnitude of the new-word problem, not only do we have to understand how prevalent they are, but we also have to understand their effects on a continuous-speech recognizer (that does not already have the capability to detect them).[9]

4. *Examine the issues involved in the new-word learning problem.* Since there are several components of a recognition system that may need updating when incorporating a new word into the working vocabulary, we wish to understand which require the most attention. The ultimate goal is to build systems that make learning new words easy, perhaps even automatic.

[9] We are interested in the occurrence of new words in *continuous* speech, where word boundaries are not readily apparent. We feel that the new-word problem is qualitatively different in *isolated-word* speech, except in terms of language modeling.

1.4 Thesis Overview

This thesis is divided into six chapters. In Chapter 1 we introduce the problem of new, out-of-vocabulary words, describe the speech recognition/understanding problem in general, discuss prior and related research, and outline the goals of this thesis.

In Chapter 2 we study the general problem of new words by examining a wide variety of corpora ranging from spontaneous speech collected during human-machine interaction to large-vocabulary newspaper texts containing literally millions of sentences. We study corpora from three different languages in an attempt to see if the general characteristics are language-independent. We examine issues such as vocabulary growth and frequency of new words, and we try to characterize new words in terms of their syntactic and phonological properties. This study is at the word level and is recognizer-independent.

In Chapter 3 we describe the recognizer we will use throughout the rest of the thesis. This recognizer is the SUMMIT system, which is different from most current systems in that it is segmental instead of frame-based (e.g., HMMs). We also discuss how it can generate N-best lists of utterance hypotheses, and the problems associated with them.

In Chapter 4 we present a novel algorithm for computing word graphs. These word graphs are more efficient than N-best lists in terms of computation time and representation space, yet they contain the very same N hypotheses. In order to study the interaction of the recognizer with new words, we find that word graphs are convenient because they can represent a very deep recognition search. Further, we will introduce some exploratory data analysis tools that are based on information contained in the graphs.

In Chapter 5 we study the new-word problem in the context of recognition. We try to characterize what happens when a recognizer encounters a new word for which it has no new-word modeling or detection capability. It is important to know how badly new words affect performance and to understand what kinds of errors they introduce. We also investigate some of the issues related to learning new words by studying the importance of learning within the different recognizer components.

Finally, we summarize the findings of this thesis in Chapter 6 and discuss the implications for solving the new-word problem.

Chapter 2

A Lexical, Phonological, and Linguistic Study

In this chapter we present an empirical study of the magnitude and nature of the new-word problem. The study is general in that we examine new words in several different corpora, spanning different types of tasks and languages. We examine issues such as vocabulary size and rate of new-word occurrence versus training set size. We find that the rate of new words falls with increasing vocabulary size, but that it does not reach zero even for very large training sets and vocabulary sizes. Therefore, speech systems will encounter new words despite the use of massive vocabularies. Having demonstrated that new words occur at all vocabulary sizes, we proceed to characterize new words and their uses. This study is at the *text* or *orthographic* level, and therefore is independent of any speech recognition/understanding system. We also examine multiple languages in an attempt to generalize across languages.

2.1 Methodology

Because the very definition of a new word depends on a system vocabulary, we must address the issue of *vocabulary determination* before we can even study the new-word problem. Vocabularies can be built by hand, automatically, or by some combination of the two. In any case, the notion of *training* data is important for determining system vocabularies.

Training data can include sample utterances/sentences or a database relevant to the task. Often, the development of a speech recognition/understanding system involves training on a large set of utterances. Not only are these utterances used to train acoustic models, but they are also used to determine a vocabulary by observing word frequencies. There may also be additional training material available for the task at hand. For example, to build a telephone directory assistance system, we would make use of phone books for the geographical area of interest. While such databases may not help with acoustic or language modeling, they can be invaluable in setting up a system vocabulary.

Of course, the most general training source for vocabulary determination is a machine-readable dictionary. However, there are two problems with most dictionaries: they are too large, and they do not contain word frequencies. Most speech recognition systems today do not have vocabularies as large as 40,000 words, and even if they can, recognition with such large vocabularies requires a great deal of computation. Generally, we would like to select a subset of the dictionary words that are relevant to the task at hand. Since most dictionaries do not have word-frequency information, we cannot select the most likely words. Even if they do, the frequencies may not be appropriate for the desired recognition task. Therefore, we generally need task-specific data.

In addition to training sets, we also use independent *testing* sets to evaluate vocabulary coverage. After all, there is no guarantee that vocabularies built from training data will fully cover all the words in a test set. There are three primary reasons why words may be missing from a vocabulary:

1. There may be a mismatch between training and testing data. The training data may be from a different task or may be too general.

2. There may not be enough training data. We can think of the process of collecting training data as random selection with a hidden, underlying distribution over all possible words. If we do not have enough training data we will miss words simply due to chance; we are most likely to miss low-frequency words.

3. Words can be invented, particularly words in open classes such as names. We

> cannot expect a training set to do more than capture the most prevalent names and recently invented words. With the invention of words, a task's vocabulary may be time-dependent.

We examine these issues in our experiments of Section 2.3.

Because we want this study to be as general as possible, we examine multiple corpora, including different tasks and languages. The tasks range from human-computer problem solving with relatively small vocabularies to newspaper dictation with extremely large vocabularies. The languages are English, Italian, and French, but most of the corpora are English. Given the large variety of tasks, we feel that we are able to reach some general conclusions regarding aspects of the new-word problem.

Some of our corpora contain speech utterances with orthographic (text) transcriptions, and some contain only text. For the speech corpora, we ignore the speech signals and examine only the orthographic transcriptions. With speech input, particularly with spontaneous speech, there is the complication of spontaneous speech events including filled pauses (e.g., "um" and "uh") and partial words. For this study we chose to discard spontaneous speech events altogether and examine only fully formed words.

One question we ask is, how do vocabularies grow as the size of the training set increases? If a vocabulary continues to grow, then new words are likely. After all, the vocabulary increases because we *continue* to find new unique words during training. Had we been using a smaller training set, these words would have been new, out-of-vocabulary words. On the other hand, if the size of a vocabulary levels off, we do not expect many new words, provided that the testing conditions are similar to the training conditions. We examine this issue in Section 2.4.

While vocabulary growth characteristics give us some indirect evidence of the likelihood of new words, they do not measure the likelihood explicitly. In Section 2.5 we measure vocabulary coverages, and thus the new-word rates, for varying training set and vocabulary sizes.

It is important to understand the effects that task, language, and training set size have on vocabulary growth and new-word rates when the training and testing tasks are the same. However, there may be times when we are interested in porting a vocabulary to a slightly different task. For example, having built a telephone directory assistance system for the Boston area, we may want to use the system in New York City. How well will the original vocabulary work in the new task? In other words, how portable is the vocabulary to another (related) task. If vocabularies are very portable, we should be able to change tasks without a dramatic increase in the rate of new words. We examine this issue in Section 2.6.

Finally, we examine some important properties of new words so as eventually to be able to model them within a speech recognition/understanding system. Since the set of names is essentially infinite, we would expect a large fraction of new words to be names. Are new words mostly names? If they are not just names, what are they? In Section 2.7 we examine the usage of new words in terms of parts-of-speech, where one of the parts-of-speech is the proper noun (i.e., name). Further, we might expect new words to be longer than more frequent and in-vocabulary words. We examine the phonological properties of number of syllables and number of phonemes for new words in Section 2.8.

2.2 Corpora

We examined the orthographic transcriptions of nine corpora in our experiments. These corpora differ in several respects including task, speech versus text, open versus closed vocabulary, intended for human audience versus a spoken language system, language, sentence complexity, and size. Some of these differences are summarized in Table 2-1. The following corpora were used for our experiments: ATIS, BREF, CITRON, F-ATIS, I-VOYAGER, NYT, SWITCHBOARD, VOYAGER, and WSJ. Examples of utterances from each of the corpora are listed in Table 2-2.

The ATIS and F-ATIS corpora consist of spontaneous speech utterances collected interactively for ARPA's Air Travel Information Service (ATIS) common task and are in English and French, respectively. The ATIS corpus contains utterances from both the so-called ATIS-2 and ATIS-3 sets [17, 29, 53]. ATIS-3 represents an increase in the

Corpus	Language	Type	Total Words	Words/ Sentence
ATIS	English	spontaneous human/computer interactive problem solving	258,137	9.8
BREF	French	read newspaper text	61,850	16.5
CITRON	English	spontaneous human/human directory assistance request	92,774	5.3
F-ATIS	French	spontaneous human/computer interactive problem solving	9,951	9.8
I-VOYAGER	Italian	spontaneous human/computer interactive problem solving	9,380	10.1
NYT	English	newspaper text	1,659,374	—
SWITCHBOARD	English	spontaneous human/human conversation	2,927,340	8.1
VOYAGER	English	spontaneous human/computer interactive problem solving	35,073	8.1
WSJ	English	newspaper text	37,243,295	22.8

Table 2-1: Corpora used in experiments. For the speech corpora, the orthographic transcriptions were processed to remove disfluencies due to spontaneous speech. For the text corpora, punctuation was removed, hyphenated words were separated, and numerals (e.g., "1,234") were collapsed to "0". (The words/sentence value for NYT is missing because the raw newswire data was not parsed into sentences.)

Corpus	Example Utterance/Sentence
ATIS	I would like a morning flight from Philadelphia to Dallas with a layover in Atlanta.
BREF	Il était debout, marchant de long en large, la caméra tentant de le suivre comme un ballon de football changeant sans cesse d'aile.
CITRON	West Coast Videos in Revere on Broadway please.
F-ATIS	Je veux arriver en fin de matinée à Dallas.
I-VOYAGER	Come faccio ad arrivare a la Groceria da Central Square?
NYT	Just before my driveway is a sweeping blind curve, so following drivers cannot anticipate the reason for my turn signal.
SWITCHBOARD	Uh, carrying guns are going to be be [*sic*] the people who are going to kill you anyway.
VOYAGER	Could you tell me how to get to Central Square from five fifty Memorial Drive, please?
WSJ	An index arbitrage trade is never executed unless there is sufficient difference between the markets in New York and Chicago to cover all transaction costs.

Table 2-2: Example utterances/sentences from the corpora.

vocabulary size, primarily due to a larger number of cities and airports.[1] In contrast, F-ATIS [10] includes only those cities and airports that are a part of ATIS-2. Utterances were collected from users trying to solve travel planning problems through interaction with a spoken language system. For some of the utterances an actual speech recognition system was employed; for others, a human "wizard" was used to perform the actual speech recognition. We used orthographic transcriptions of the utterances with spontaneous speech disfluencies removed.

The VOYAGER corpus consists of spontaneous speech utterances in English collected interactively for the MIT VOYAGER urban navigation and exploration system [76]. The I-VOYAGER corpus is similar, except that the utterances are in Italian [22]. Utterances were again collected using a human "wizard." Again, we removed spontaneous speech disfluencies from the orthographic transcriptions.

The CITRON corpus consists of utterances collected by NYNEX from actual directory assistance telephone calls [12, 68]. The users interacted with human operators.

The SWITCHBOARD corpus consists of spontaneous human/human dialogs collected by Texas Instruments [25]. The dialogs are based on a large set of predefined topics. The topics were selected to be of general interest and to encourage active discussion. We include both sides of dialogs in our study. We removed spontaneous speech disfluencies from the orthographic transcriptions.

The WSJ [55] and NYT corpora consist of English text from the *Wall Street Journal* and the *New York Times* newspapers, respectively. The text for WSJ was made available by the ACL Data Collection Initiative [6] and represents three years (1987–1989) of newspaper text. The text for NYT was collected over a period of three months in early 1994 via a newswire service.

The BREF corpus consists of read utterances collected by LIMSI-CNRS [24,39]. The sentences, in French, were selected from three months of the newspaper *Le Monde.* The selection of sentences explicitly maximized the number of phonemic contexts and the number of distinct words. This selection process was not random and therefore could bias the vocabulary growth and new-word rate characteristics of this corpus.

[1] We describe the distinction between ATIS-2 and ATIS-3 in more detail later in Section 5.2.1.

2.3 Data Preparation and Vocabulary Determination

Even though some of these corpora were collected as speech, we used only their orthographic transcriptions for the experiments in this chapter. Because the speech utterances were spontaneous they contained disfluencies, some of them resulting in partial words. To keep the effort required for this thesis manageable, we deleted all partial words from the transcriptions. The subject of partial words is certainly related to the new-word problem, but we feel that it is beyond the scope of this study.

The text and orthographic transcriptions required further preparation, regarding capitalization, numerals, and punctuation. We converted all text to lowercase because in speech recognition, case distinctions are meaningless. Because the set of numerals is infinite, we collapsed all numerals that were not spelled out (i.e., strings of digits) down to "0". This is reasonable since, when such digit strings are actually spoken, only a relatively small vocabulary is required.

As far as punctuation is concerned, we removed all of it except for the apostrophe. In English, we left possessives and contractions alone. For example, "wouldn't" and "Alexander's" were left as is. However, in French and Italian, the elision that occurs when words are joined by apostrophe would account for a large growth in the number of distinct words. We felt that this type of word combination was much more of a problem than contractions and possessives in English. Therefore, for French and Italian sentences we decided to break words apart at apostrophes. For example, "l'époque" became "l' époque" (two words). For all languages, we broke hyphenated words apart, again because we felt that they caused an artificially large number of words. This meant that "new-word rate" became "new word rate". In terms of speech input, the two would be indistinguishable.

Of course, with such large corpora there are bound to be spelling errors, but we did not attempt to correct them. We deemed it to require too much effort to locate and correct them. Random sampling of the singleton words in our corpora indicated that spelling errors did occur, but were not a significant problem.

We performed several experiments to try to understand the phenomena of new words. Because a new word is defined as an out-of-vocabulary word, it is important to understand what we considered to be a word, as well as how we determined vocabularies. We defined a *word* to be a string of characters delimited by spaces after performing the aforementioned preprocessing. We determined vocabularies automatically by observing a set of text, the training set, and placed all words that occur at least n times in the vocabulary. That is, we defined the vocabulary to be the set of words $\mathcal{V}$ such that $\mathcal{V} = \{w : c(w) \geq n\}$, where $c(w)$ is the observed count of word w in the training set. For our experiments, typically $n = 1$, meaning that our vocabulary consisted of all unique words in the training set.

We admit that this is a simplistic definition of words[2] and a simplistic method of building vocabularies. One obvious flaw with our vocabulary-building paradigm is that it does not guarantee completion of closed sets of words (e.g., days of the week). However, in the interest of expediting experiments involving millions of words, we decided to adopt this simple but slightly flawed approach because we think it is an adequate model of empirical vocabulary determination.

2.4 Vocabulary Growth

Since our definition of a new word is so closely tied to a system vocabulary, we first examined the characteristics of vocabulary growth for each of our corpora. If the vocabulary size tends to level off after enough training data has been processed, then new words should not occur very frequently. If the size does not level off, we are likely to see new words.

We automatically built vocabularies by varying the quantity of training data. For a given amount of training data, we set the vocabulary $\mathcal{V}$ to be the set of all words that occur at least once. Specifically, the vocabulary size is the size of $\mathcal{V}$, $||\mathcal{V}||$, for $n = 1$.

To generate the vocabulary growth curve for each corpus, we made several passes through all of the corpus' data. For each pass, we randomized the sentence order and

[2] Clearly this definition is lacking for a language such as German in which compound words can be created arbitrarily and do not contain spaces. A better definition would be based on the morphology of the language.

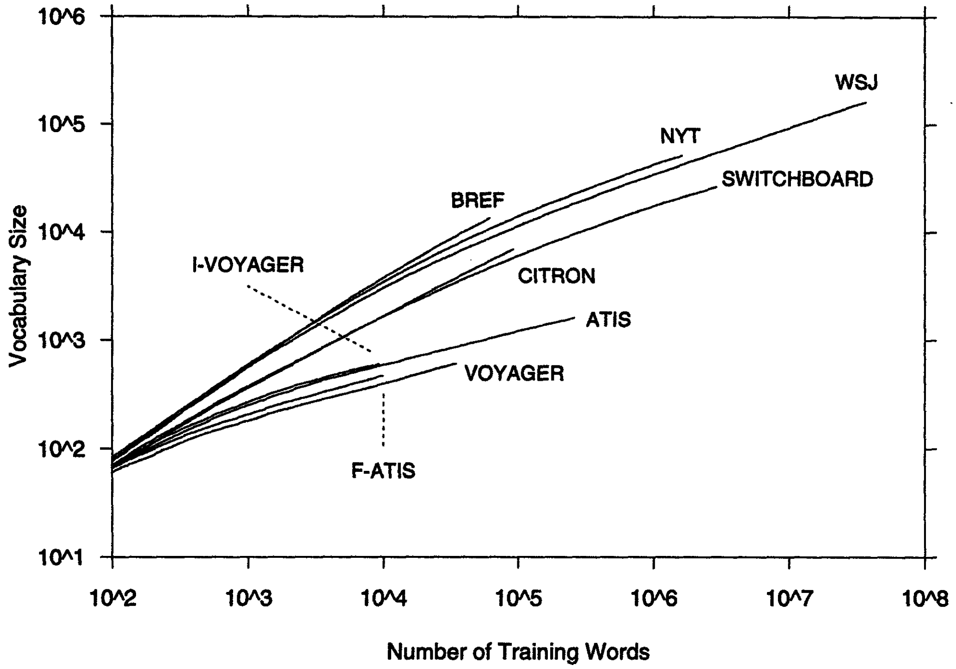


Figure 2-1: Vocabulary size versus quantity of training.

then went through the corpus, keeping track of the vocabulary size and the number of training words examined. Finally, we averaged our resulting vocabulary sizes over ten such passes to arrive at the curves displayed in Figure 2-1.

Examining the general shape of the vocabulary growth curves, we find that the corpora cluster into two or three groups, depending on how they are interpreted. The three potential groups are:

1. ATIS, F-ATIS, VOYAGER, and I-VOYAGER;

2. CITRON and SWITCHBOARD; and

3. WSJ, NYT, and BREF.

The first group contains spontaneous utterances from interactive problem solving sessions with a speech understanding system; this group has the smallest vocabularies and the lowest rates of vocabulary growth. The second group consists of spontaneously uttered human/human communication from less limited domains. The third group consists of orthographic transcriptions of newspaper articles and has the largest vocabularies and highest growth rates. It is debatable whether groups 2 and 3 should be considered separately, and we will discuss this issue further.

The first group of ATIS, F-ATIS, VOYAGER, and I-VOYAGER form a cohesive group that is separate from the other group(s). This group contains the speech of users communicating with a spoken language system that attempts to *understand* their utterances and interacts with them, providing them with answers to their queries and asking for clarification. In the case of all four of these corpora there was an actual natural language system processing their input. (The speech was either recognized by the system, or it was transcribed by a human "wizard.") The natural language systems involved all had limited, finite vocabularies. It is reasonable to hypothesize that the limited vocabulary nature of the systems may have influenced the vocabulary used by the speakers. If a speaker used a word not in the vocabulary of a system, the system would fail to recognize and understand the utterance. In cases when a human wizard performed the speech recognition, the natural language system could explicitly notify the user of an out-of-vocabulary word by responding with something like "I don't understand the word 'Zimbabwe,' please try again." When the system was performing its own speech recognition, the user might notice the recognition errors associated with an out-of-vocabulary word. By learning of the limitations of a system's vocabulary, a user could *adapt* his or her own vocabulary to that of the system in an attempt to solve their travel or navigation problem. Since a user wants to solve a problem with the system's assistance, there is motivation to adapt queries to the limits of that system. Therefore, it is not surprising that the corpora consisting of human/computer interaction would show the smallest vocabularies and lowest growth rates. The systems' limited vocabularies may have affected the vocabularies used by the speakers during data collection. We would expect that data collected within a larger domain with a less restrictive system to display a larger vocabulary (e.g., an automated directory assistance task would likely have vocabulary size more like CITRON than ATIS).

The group of WSJ, NYT, and BREF show the highest rate of vocabulary growth. One explanation is due to the *domain* of the corpora. The newspaper texts cover wide-ranging topics; the possible topics are virtually limitless as opposed to the topics that the ATIS and VOYAGER systems are capable of handling, which are quite specific. Another explanation is that these corpora were derived from newspaper text that was not originally intended to be understood by a computer. The original intent of the text was for human/*human* communication. Because most people have relatively large vocabularies (and are even able to deduce the meaning of some words beyond their regular vocabulary based on context) the vocabulary of the newspaper text was not nearly as limited as in the case of the human/computer interactive utterances. Therefore, one explanation for the increased vocabulary is the intended audience of the text.

The third potential group consists of CITRON and SWITCHBOARD. This group consists of human/human speech communication with essentially unlimited vocabularies. It is open to debate whether or not this group is distinct from the group consisting of WSJ, NYT, and BREF. We can distinguish CITRON and SWITCHBOARD from these other corpora in that CITRON and SWITCHBOARD consist *verbal* communication. Perhaps people tend to use a larger vocabulary when they write compared to when they speak. Alternatively, we can consider these two clusters of corpora to be one. If the written versus spoken distinction is not important, this one cluster would consist of human/human communication from very broad domains. The vocabularies could be large because the corpora consist of utterances or sentences intended for human ears or eyes, without the constraint of communicating with a limited-vocabulary computer. Most of these corpora have essentially unlimited domains, except perhaps CITRON, which is broadly constrained to consist of directory assistance queries. Thus, an alternative explanation for the division into two clusters is based on limited versus unlimited task *domain*. We do not have a corpus consisting of human/human communication within a very limited domain, so we cannot readily distinguish these two alternative explanations. Although CITRON's domain is somewhat limited, the types of queries possible in directory assistance telephone calls are wide-ranging. A spoken language system operating on directory assistance–type queries might require a large vocabulary similar to that of CITRON to handle the vast number of distinct names in a typical telephone book. If this were the case, it would lend evidence to the explanation that the fundamental difference between the corpus clusters is due to domain size.

It is important to note that the clustering of the corpora appears to be *language-independent.* We find that the vocabulary growth characteristics of F-ATIS and I-VOYAGER are consistent with those of ATIS and VOYAGER despite being different languages. Further, the French newspaper text contained in BREF shows similar characteristics with that in WSJ and NYT. Although BREF has the largest vocabulary of all our corpora, higher even than WSJ and NYT, this could be due to the way utterances were selected from *Le Monde* for inclusion within BREF. As previously mentioned, the selection process explicitly maximized the number of distinct words. Nevertheless, general vocabulary size and growth rate of BREF appears to be comparable to that of WSJ and NYT. Therefore, our vocabulary growth findings appear to be language-independent.

In summary, the vocabulary growth curves support two interpretations:

- there are three groups, divided into human/computer speech interaction, human/human spoken language, and human/human written language; and
- there are two groups, divided into human/computer interaction within a limited domain and human/human interaction within an unlimited, broad domain.

The data do not allow us to readily distinguish between these two possible explanations. It is clear that there are *at least* two distinct groups: with human/computer interactive speech within a limited domain having significantly smaller vocabularies and lower growth rates than human/human speech and text within a more unlimited domain.

2.5 Vocabulary Coverage: New-Word Rate

Figure 2-1 shows us how fast a vocabulary can grow as the quantity of training data used to determine it increases, but it does not reveal how well such a vocabulary would cover unseen data. In other words, it does not give us a clear indication of the likelihood of encountering new words, or the new-word rate, for a particular type of task.

In another experiment, we attempted to estimate the rate of new words for the various corpora. Again, we made ten passes over each corpus. For each corpus in each pass, we randomly selected 15% of the corpus as a test set and set it aside. Then, we

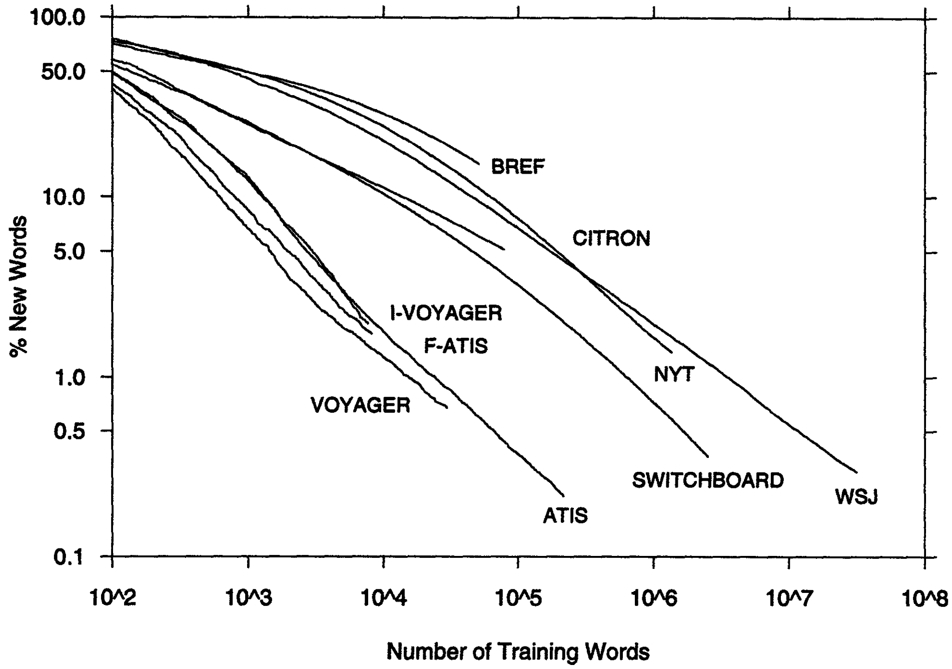


Figure 2-2: New-word rate versus quantity of training.

went through the remaining 85% of the corpus, measuring the vocabulary coverage over the test set as we built up a vocabulary incrementally. Figure 2-2 shows the probability of encountering a new word for a particular task and training set size. We estimated this probability by measuring the fraction of words in a test set that were not covered by the empirically determined vocabularies, averaged over several passes. Each curve is the *new-word rate* versus the amount of training data used to determine the corresponding vocabulary.

The clustering of the corpora is completely consistent with the clustering we observed in the previous section with respect to vocabulary growth. The corpora could cluster into two or three groups based on the size of a task's domain and/or communication a human or a machine. Again the clustering appears to be language-independent.

Figure 2-3 shows the new-word rate versus *vocabulary size* instead of amount of training data (as in Figure 2-2). In this figure, we have implicitly varied the vocabulary size by explicitly varying the training set size used to determine it (i.e., we combined the data in Figure 2-1 and Figure 2-2). This figure shows the same clustering of corpora

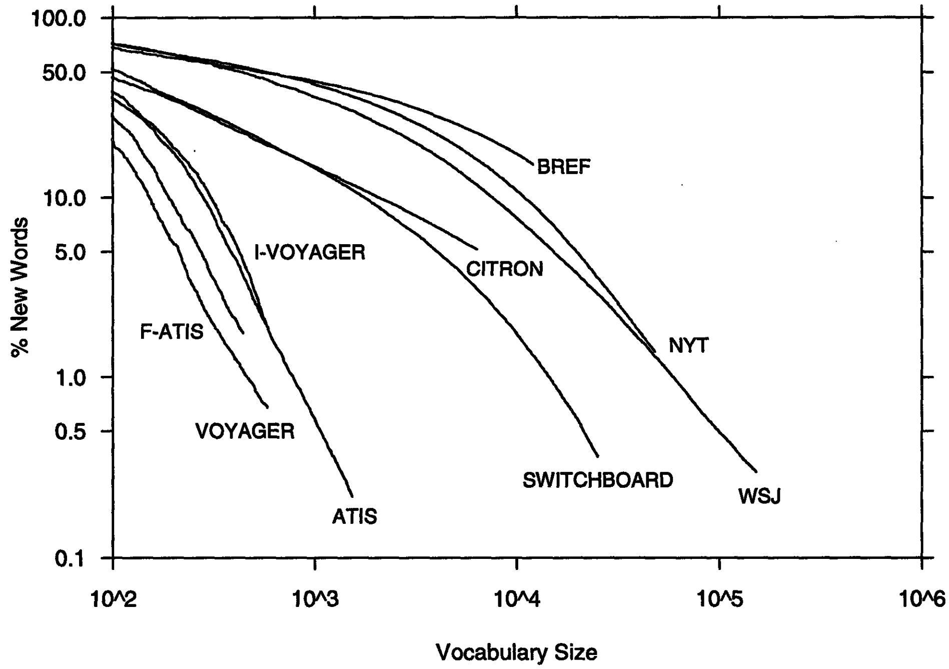


Figure 2-3: New-word rate versus vocabulary size. The vocabulary size, $v = ||\mathcal{V}||$, represents the size of the vocabulary as we increased the size of the training set. This is *not* the same as building a vocabulary of a particular size by choosing the most-frequent v words after observing all available training data.

and the same new-word rate trends.

These figures clearly show that new words will *always* be present with any reasonable training set size. The number of new words we can expect depends on the type of task and the amount of training data we use to determine a vocabulary. The dependence on the type of task is quite clear. For example, to achieve a 1% new-word rate on ATIS requires about 25,000 words of training material to determine a vocabulary of about 650 words. In contrast, to achieve the same 1% new-word rate on WSJ requires about 4,000,000 words of training material to build a vocabulary of about 65,000 words. This is roughly a two order of magnitude difference for both the amount of training data and the resulting vocabulary size.

Although a 1% new-word rate may seem low enough to be acceptable, if we measure the rate of *sentences* that contain new words, we find a much higher rate. For both the ATIS and WSJ tasks, a 1% new-word rate translates to 17% of the sentences containing at least one new word. Further, for WSJ 3.6% of the sentences contain *more than one* new word. For ATIS, the fraction of utterances containing at least two new words is 2.5%. Clearly a 1% new-word rate can imply a very large fraction of sentences containing one or more new words. A rate of nearly one in five sentences containing a new word certainly is a problem that cannot be ignored.

For some types of tasks it may be impractical to reduce the new-word rate to the point where it can be ignored. Newspaper text is particularly difficult in that the names and topics in the news tend to change with time, so even achieving an (unacceptably high) 1% new-word rate on a static set of data does not guarantee that the rate will even *remain* at that level.

Actually, we could have estimated the new-word rate directly from the vocabulary growth curves of Figure 2-1. The *slope* of each curve represents the derivative of the number of distinct words with respect to the number of total words. This is the same as the probability that the next observed word will be distinct, which is exactly what the new-word rate represents. In Figure 2-4 we plot the slope of the ATIS vocabulary growth curve as points, where the slope was estimated by a simple ratio of differences. The superimposed solid line is the explicitly computed new-word rate for ATIS from Figure 2-2. Clearly both methods of estimating the new-word rate are consistent, but the method involving estimating the slope of the vocabulary growth curve yielded a much noisier estimate, even though we were taking the slope of a smoothed (averaged) curve.

There is an alternative way to plot new-word rate versus vocabulary size based on another vocabulary determination technique. If we have a large set of training data and we want to build a vocabulary of a particular size $v = ||\mathcal{V}||$, we would likely compute word frequencies over *all* of the training data and put the v most-frequent words in the vocabulary. This is different from the way we have been determining vocabularies up to this point. It assumes that we have all the training data ahead of time with which to build our v-word vocabulary. We would expect this technique to yield a lower new-word rate since we have access to more word-frequency information for vocabulary determination. Figure 2-5 compares the new-word rate computed using both methods for the WSJ corpus. We do indeed see lower new-word rates when building a vocabulary

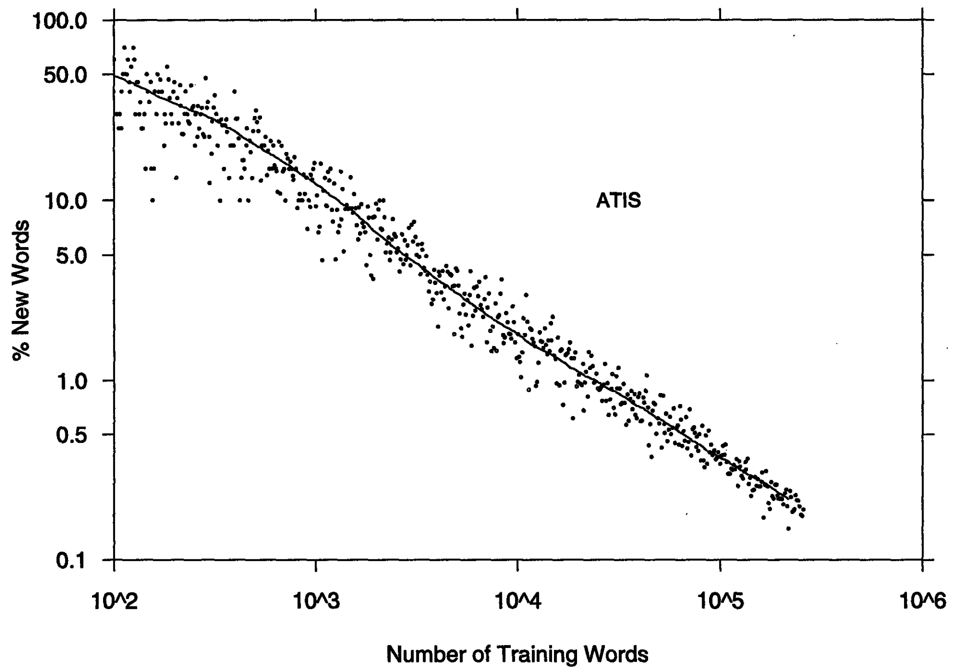


Figure 2-4: New-word rate as slope of vocabulary growth. The points represent the slope along the vocabulary growth curve of Figure 2-1 for ATIS. The line is the explicitly computed new-word rate from Figure 2-2.

of a particular size v using all of the training data, with the biggest reduction of new-word rate coming when v is relatively small. As v approaches its maximum size, the two methods of vocabulary determination converge.[3]

Therefore the new-word rate curves of Figure 2-3, unlike those of Figure 2-2, are implicitly dependent on the amount of training data available for each corpus. For this reason we chose not to fit a function to them.

2.6 Vocabulary Portability

As we have seen, determining large vocabularies can require very large amounts of training data for some types of tasks. Do we always need such large quantities of training material to determine vocabularies, or can we use a vocabulary from one task

[3] Actually, the two methods should yield the same vocabulary, and hence the same new-word rate, when all of the words in the training set are used (at the rightmost points on the curves). However, because of our random sampling of testing sets we see a small difference between the curves at maximum vocabulary size.

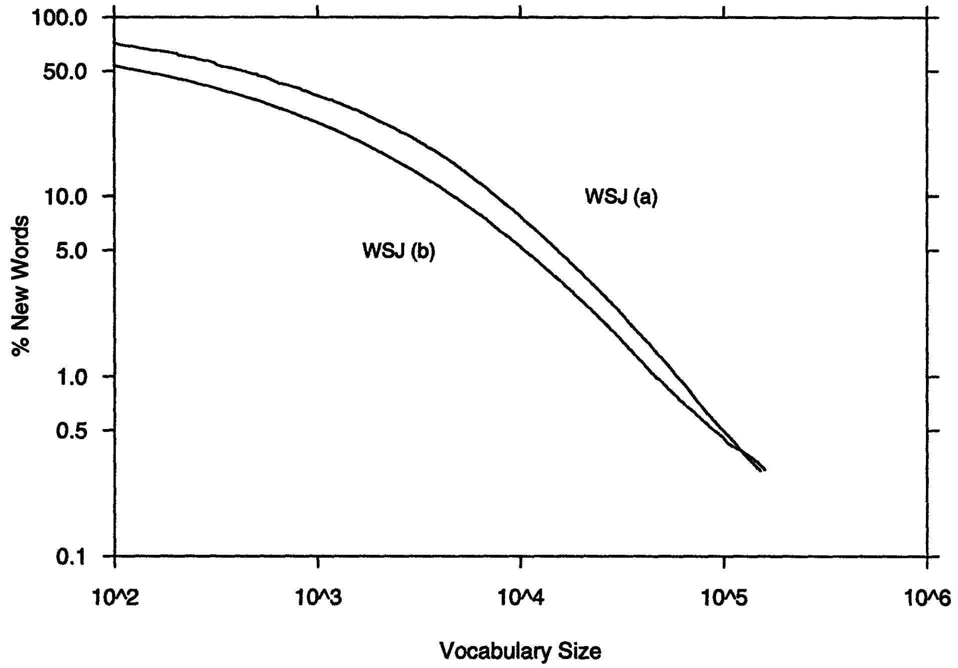


Figure 2-5: New-word rate for different methods of vocabulary determination. This plot demonstrates the difference between two vocabulary-determination methods using the WSJ corpus. For (a) we varied the training set size and set the vocabulary to include all unique words, implicitly changing the vocabulary size. For (b) we used the *entire* training set to compute word frequencies and explicitly varied the vocabulary size $v = ||\mathcal{V}||$ by setting the vocabulary to include only the v most-frequent words.

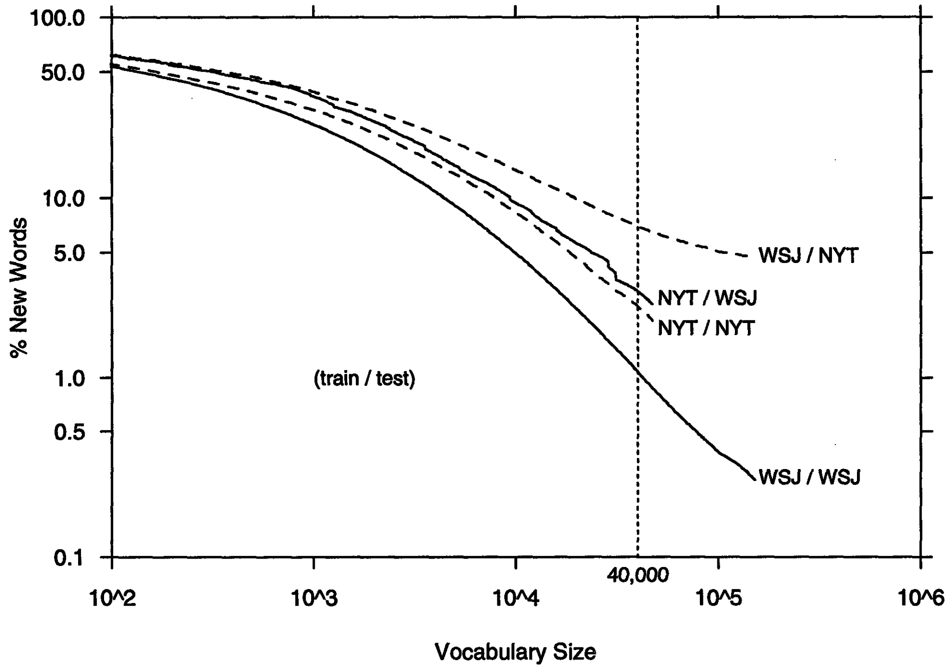


Figure 2-6: Coverage across WSJ and NYT. New-word rate for training and testing on all combinations of WSJ and NYT. Vocabularies are built by observing the entire training set and adding words in decreasing order of frequency.

for another (related) task? We would hope that a large fraction of one vocabulary could be useful for another task.

In an attempt to gauge vocabulary portability for similar tasks, we experimented with the WSJ and NYT corpora since both contain edited newspaper text, and both can produce large vocabularies. Figure 2-6 shows the within-task and across-task new-word rate curves for these two corpora. Because we assumed we had all training material for a particular task before evaluating vocabulary coverage on another task, we chose to build the vocabularies in decreasing order of word frequency as we discussed in Section 2.5. We divided each corpus into a fixed test set (15%) and a fixed training set (85%).

The two curves with the lowest new-word rates are, not surprisingly, the curves for task-dependent vocabularies. To examine the portability of the WSJ vocabulary to the NYT task, compare the two dashed lines. The difference between the curves indicates the increase in new-word rate when task-dependent data is not available. Likewise, to examine the portability of the NYT vocabulary to the WSJ task, compare the two solid

Training	Testing	r
WSJ	WSJ	1.08%
NYT	WSJ	3.04%
NYT	NYT	2.52%
WSJ	NYT	6.95%

Table 2-3: Cross-task new-word rates for WSJ and NYT. Vocabularies are set to contain the 40,000 most-frequent words in the training corpus. The new-word rate r is measured over the entire testing corpus.

lines. Table 2-3 summarizes these portability curves at the 40,000-word vocabulary size.

On the WSJ test set, we see an increase in the new-word rate by a factor of 2.8 at the 40,000-word vocabulary size when we do not use a task-dependent vocabulary. Similarly, on the NYT test set, the new-word rate also increases by a factor of 2.8. However, the flattening out of the WSJ/NYT curve suggests a problem of diminishing returns with non-task-dependent vocabularies. Apparently, very low-frequency words in one task are not very useful in reducing the new-word rate in another task. Presumably, these low-frequency words are more task-dependent than higher-frequency words.

Although the WSJ and NYT corpora are similar in that they are both edited newspaper texts, there are some important differences. First, the topics covered in NYT text tend to be more general in nature, whereas the topics in WSJ are largely business and financial in nature. This is evident in Table 2-3, in which the new-word rates measured on NYT test material are higher than those measured on WSJ. Second, the two corpora were collected over different time periods. Names and topics in the news tend to evolve with time so we can expect some time-dependency in the vocabularies. Because our WSJ and NYT data were collected years apart, 1987–1989 and 1994 respectively, we attempted to estimate how time-dependency affected our WSJ/NYT cross-task coverages.

We decided to use WSJ to examine time-dependency of vocabularies because we have a large amount of WSJ data spanning three years. For this study, we divided WSJ into four pieces: WSJ-87 from 1987, WSJ-88 from 1988, WSJ-89 from late 1989, and WSJ-test, the test set, from early 1989. This division gives us data before and after the test set. In addition, WSJ-all contains all WSJ data not in the test set.

Table 2-4 summarizes our results for vocabularies containing the 40,000 most-

Subset	Time Span	Size (words)	r
WSJ-87	all 1987	17,283,667	1.38%
WSJ-88	all 1988	14,495,972	1.34%
WSJ-89	late 1989	4,649,493	1.38%
WSJ-all	1987, 1988, late 1989	36,429,132	1.19%
WSJ-test	early 1989	814,163	—

Table 2-4: Time-dependency of vocabularies in WSJ. The new-word rate r for each training set is measured over the same testing set, WSJ-test, for a vocabulary containing the 40,000 most-frequent words from each training set.

frequent words. Surprisingly, we found no significant difference in vocabulary coverage as a function of time. Additionally, we found no real differences even as we used vocabularies larger than 100,000 words. The lowest new-word rate was achieved by building a vocabulary using WSJ-all. Presumably, using more data allows more accurate measurement of word frequency, especially for the low-frequency words, resulting in better vocabulary selection. Even though WSJ-89 was the smallest training set, it achieved a new-word rate comparable to the much larger WSJ-87 and WSJ-88 probably because it was collected *after* WSJ-test. This fact might indicate some time-dependence. Perhaps we would see more time-dependence in vocabulary coverage if we could examine WSJ data spanning a larger timer period than three years. The surprising result of very little time-dependence of WSJ vocabularies leads us to believe that the most significant difference between the WSJ and NYT data is the difference of topics and not the time at which the data were collected.

In summary, this limited study of vocabulary portability tells us a couple of important things:

- porting a vocabulary from one task to another, similar task can result in a significant elevation of the new-word rate; and
- low-frequency words appear to be particularly task-dependent.

We found that porting a fairly large (40,000-word) vocabulary from the WSJ to the NYT task, and vice versa, results in nearly a three-fold increase in new-word rate. Even though the WSJ and NYT corpora are similar in that they both consist of English newspaper text, large vocabularies determined on one of them suffer from increased numbers of new words on the other. We hypothesized that part of the explanation was due to the time-dependent nature of topics in the news. However, we examined vocabularies and their coverages within the three years of the WSJ data and did not find significant time-dependence. However, the four years separating the collection of the WSJ and NYT data may contribute to some time-dependence of news topics. A more likely explanation is that the topics of NYT are more general in nature compared to the primarily business and financial topics of WSJ. This explanation is supported by the fact that within-task and across-task new-word rates measured on NYT test data were more than twice as high as the rates measured on WSJ (see Table 2-3). If we were to examine more dissimilar tasks, we would expect even worse across-task vocabulary coverage.

We found that relatively low-frequency words were more task-dependent. Figure 2-6 shows that the slopes of the *across-task* curves for new-word rate are flatter than the slopes of the *within-task* curves at larger vocabulary sizes, indicating that the lower-frequency words from one task are less helpful in reducing the new-word rate on another task than they are on their own task. This implies that these words are relatively task-dependent. What does this mean for porting vocabularies from one task to another? It means that the relatively high-frequency words will be most useful. Unfortunately, those same high-frequency words are the words most easily determined when empirically building a vocabulary with task-dependent data. It is the *low-frequency* words that reduce the new-word rate to much below 5–10%, and it is these words that are ***most difficult to determine*** empirically. After all, because they are relatively infrequent, large amounts of task-dependent training text is required just to identify them.

We have performed a very limited study of vocabulary portability. To really verify our hypotheses that across-task new-word rates are significantly higher than within-task new-word rates and that low-frequency words are the words most task-dependent, we should examine the problem of porting vocabularies on more tasks. However, we feel that such a study is beyond the scope of this thesis. The limited vocabulary portability experiments we carried out in this section give us a general idea of the problems associated with porting vocabularies to other, similar tasks.

2.7 New-Word Part-of-Speech

Given that encountering new words is inevitable, we wanted to characterize their usage so as ultimately to develop better language models to accommodate them. Because we chose a straightforward word-frequency approach to building vocabularies, it is natural to consider low-frequency words to be potential new words. Words that occur only once in a corpus are the words most likely to be missed when building a vocabulary empirically.

For our analysis of new-word usage we chose to examine syntactic part-of-speech tags. We collapsed a large set of forty-eight tags [16] down to eleven: proper nouns, nouns, adjectives, adverbs, verbs, conjunctions, pronouns, numbers, determiners, prepositions, and "other." One aspect of the new-word problem that we were particularly interested in examining was the fraction of new words that are names (i.e., proper nouns). Because the set of names is essentially infinite, it is commonly hypothesized that most new words are names. Our choice of part-of-speech tags included a proper noun tag, so we were able to evaluate this hypothesis.

We chose to examine parts-of-speech of potential new words using the WSJ and ATIS corpora. For WSJ, we used the 57,712 hand-tagged sentences that came with the corpus. For ATIS, we automatically tagged the entire corpus of 26,583 utterances. We performed the tagging using Brill's part-of-speech tagger [11] trained using nearly 1,800 hand-tagged utterances. For ATIS, we corrected by hand the tags of the words that occurred only once. For the purposes of part-of-speech analysis, we set the vocabulary for each task to include all the words that occurred at least twice (i.e., $n = 2$). The remaining words that occurred only once, the singletons, were used as simulated new words. For WSJ this procedure yielded a vocabulary size of 36,582 words and 24,348 new words (that occurred only once each). For ATIS, it resulted in a vocabulary size of 1,152 words and 457 new words.

Table 2-5 displays the part-of-speech distributions for WSJ and ATIS. Two distributions are given for each corpus: one for the in-vocabulary words and a second for the simulated new words. The distributions for the in-vocabulary words are *unweighted* by

	WSJ		ATIS	
part-of-speech	V	N	V	N
proper noun	31.5	32.9	18.2	10.5
noun	31.0	28.3	37.4	48.6
verb	17.9	10.7	20.3	24.9
adjective	16.1	25.4	10.1	9.6
adverb	2.6	1.9	4.2	3.7
number	0.2	0.1	2.5	0.0
conjunction	<0.1	<0.1	0.3	0.0
determiner	<0.1	<0.1	1.5	0.0
preposition	0.2	<0.1	2.5	0.0
pronoun	0.1	<0.1	1.2	0.2
other	0.4	0.6	1.6	2.4

Table 2-5: Part-of-speech distributions for WSJ and ATIS. All values are percentages. *V* indicates over all in-vocabulary words, unweighted by word frequency. *N* indicates over all simulated new words.

word frequencies. For these two corpora, potential new words are largely nouns, proper nouns, verbs, and adjectives. Further, we see that proper nouns, or names, do *not* make up the vast majority of potential new words, but instead are roughly comparable to nouns and verbs in terms of percentages. Thus, new words are not dominated by names.[4]

Further, function words are almost completely unrepresented in the set of simulated new words. From the standpoint of new-word detection, it is fortunate that new function words are very unlikely. It is well-known that function words tend to be short and poorly articulated. Detecting such words as new words would likely be very difficult based on the relatively poor acoustic evidence they would provide.

Not only are function words relatively high-frequency words, there are not very many of them. We examined a large machine-readable dictionary (*Moby Part-of-Speech 1.3*) annotated with part-of-speech tags. We found that out of well over 200,000 words listed, only 320 were tagged as possible function words (conjunctions, prepositions, pronouns, definite articles, and indefinite articles in the list of available tags).[5] This list

[4] Cursory examination of CITRON's potential new words reveals that more than 60% are names. Thus, the fraction of new words that are names is task-dependent. We were not able to perform a full part-of-speech analysis of CITRON because we did not have hand-tagged CITRON text with which to train Brill's part-of-speech tagger.

[5] Examining the hand-tagged WSJ data, we found that about 200 function words occurred.

of function words is short enough that we could include all of them if we were building a task-independent vocabulary. Further, the list of function words appears to be task-independent in nature, meaning that one (large) list of function words is likely to cover the function words in a wide range of tasks. Thus, we do not really have to worry about new function words.

As we mentioned in Section 1.2.1, Suhm et al. [69] similarly examined new words in the WSJ corpus. They found that 27% were names, which is comparable to our 33%. The difference may be due to the fact that our analysis was performed on more data, with more (simulated) new-word occurrences.

2.8 New-Word Phonological Properties

Because we are ultimately interested in detecting new words, we wanted to examine some of their phonological properties. The two properties that we examined were the number of syllables and the number of phonemes per word.

We examined WSJ and ATIS in detail, using all of both corpora. In order to ascertain the number of syllables and phonemes in each word, we looked them up in a large on-line dictionary (*Moby Pronunciator 1.3*) containing over 167,000 entries. For our comparison between in-vocabulary and out-of-vocabulary words, we divided each of the WSJ and ATIS vocabularies into two subsets, with the in-vocabulary words being all the words that occurred at least twice. For WSJ, the vocabulary size was 107,101 words, leaving 55,246 new words. For ATIS, the vocabulary contained 1,127 words, leaving 489 new words.

Figure 2-7 shows distributions for number of syllables per word. Again, two distributions are given for each corpus: one for the in-vocabulary unweighted by word frequency and the second for the set of simulated new words. Table 2-7 shows the mean for each condition. The distributions show that on average (unweighted) new words are slightly longer than in-vocabulary words by about 0.2–0.3 syllables. (In general, words in ATIS tend to be shorter than words in WSJ. Perhaps people tend to use longer words when they write.) Table 2-6 summarizes our findings for number of phonemes per

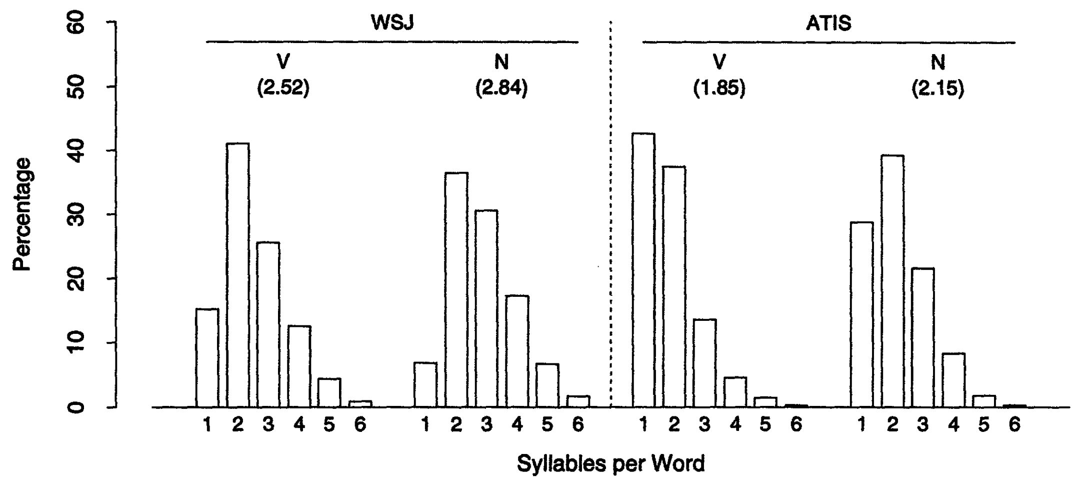


Figure 2-7: Distributions for number of syllables per word for WSJ and ATIS. *V* indicates over all in-vocabulary words, unweighted by word frequency. *N* indicates over all simulated new words.

	WSJ		ATIS	
	V	N	V	N
mean number of syllables	2.52	2.84	1.95	2.15
mean number of phonemes	6.58	7.23	5.01	5.62

Table 2-6: Mean number of syllables and phonemes per word for WSJ and ATIS. *V* indicates over all in-vocabulary words, unweighted by word frequency. *N* indicates over all simulated new words.

word. Because the same trends visible in the number of syllables distributions appear in the number of phonemes distributions, we have displayed only the mean number of phonemes per word.

In general, new words are slightly longer than in-vocabulary words, even when the in-vocabulary words are not weighted by word-frequency. On average, they are about 0.3 syllables (13%) and 0.6 phonemes (10%) longer. Suhm et al. [69] found almost no difference between the distributions of number of phonemes in a similar study on the WSJ corpus. This discrepancy may be due to the fact that we examined a much greater quantity of data and used much larger vocabularies than Suhm et al. did in their study. However, we find only relatively small length differences between new words and in-vocabulary words, in terms of both number of syllables and number of phonemes. The fact that new words tend to have more syllables and phonemes suggests that they may be slightly easier to detect than shorter words, but the difference is very small and may not have much impact on new-word modeling.

2.9 Summary

In this chapter we have examined important aspects of the new-word problem at the lexical, linguistic, and phonological levels. We performed our study using a wide range of corpora spanning different types of tasks and languages in an attempt to study the new-word problem in a general manner.

We first demonstrated the vocabulary growth characteristics for several corpora. Naturally, the less restrained tasks show the largest vocabularies and highest vocabulary growth rates. However, the real measure of the new-word problem is the rate of new words. We demonstrated that although the new-word rate drops with increasing training set and vocabulary size, it does not reach zero. In fact, it can take very large vocabularies, on the order of 100,000 words or more, even to get the new-word rate down to 1% for some types of tasks. We showed that although a new-word rate of 1% may seem low enough, it can correspond to 17% of sentences containing one or more new words. Having nearly one in five utterances containing a new word is almost certainly an unacceptably high rate. Because of the misrecognition and misunderstanding that a new word could cause, a sentence rate as high as one in five would likely interfere with a user's interaction with a spoken language system. When new words *do* occur, they cannot simply be ignored without compromising the usefulness of a system.

We found that the vocabulary growth and coverage characteristics of our corpora allowed them to be clustered into at least two distinct groups. Further, the clustering of the corpora was completely independent of their languages. There were three possible factors affecting the clustering:

1. limited versus unlimited task domain;
2. mode of communication (i.e., spoken versus written); and
3. intended audience (i.e., another human or a machine).

The four corpora having the smallest vocabularies and lowest new-word rates (ATIS, F-ATIS, VOYAGER, and I-VOYAGER) are all limited domain, spoken, and intended for a spoken language system. The three corpora having the largest vocabularies and the highest new-word rates (WSJ, NYT, and BREF) are all unlimited domain, written, and intended for human eyes. The two corpora with intermediate vocabulary sizes and new-word rates (CITRON and SWITCHBOARD) are both relatively unlimited domain, spoken, and directed at another human.

Given that new words do occur to some extent no matter the vocabulary size, we examined some of their characteristics. We studied the usage of new words by examining syntactic parts-of-speech. We found that new words are largely nouns, proper nouns (i.e., names), adjectives, and verbs; the majority of them are not necessarily names. Further, we examined the length of new words, as measured by the numbers of syllables and phonemes per word, and found that they tend to be only slightly longer than in-vocabulary words, both weighted and unweighted by word-frequency. Hopefully the knowledge gained in terms of new-word part-of-speech and length distributions will be helpful in modeling new words phonologically and linguistically.

One of the primary goals of this thesis is to demonstrate that new words are a real problem for a wide range of speech recognition/understanding tasks. We hope this chapter has done just that. We also hope that this work encourages others to address the new-word problem.

Chapter 3

SUMMIT System

In our work on word graphs in Chapter 4 and our new-word recognition experiments in Chapter 5 we make extensive use of a continuous-speech recognition system. In this chapter we briefly describe SUMMIT, the continuous-speech recognition system developed by the Spoken Language Systems Group of the MIT Laboratory of Computer Science.

The SUMMIT speech recognition system [56, 66, 72, 74–77] is different from most other systems in that it is segment-based instead of frame-based. Most systems today utilize hidden Markov models (HMMs) to model acoustic features measured over a sequence of fixed-rate frames.[1] These frames are usually very short in duration, typically 10 ms. Since this duration is much shorter than most individual phonetic units, HMMs model the phonetic units as sequences of frames. In contrast, the SUMMIT system initially proposes a set of variable-length segments that are generally intended to correspond to individual phonetic units. The rationale is that modeling entire phonetic units is superior to modeling small, fixed-length frames. One reason is that many acoustic measurements for phonetic discrimination are at the segmental level. Another reason is that the HMM framework makes the assumption that measurements from the individual frames are statistically independent, which is clearly invalid for relatively steady-state phonetic units (e.g., long vowels). SUMMIT makes a similar assumption that its seg-

[1] However, the use of HMMs does not necessarily imply fixed-rate frames.

mental measurements are statistically independent, but because entire phonetic units are modeled, this assumption seems less severe.

The SUMMIT system is not the only segmental system. Other segmental systems include the Stochastic Explicit-Segment Model of Leung et al. [40, 41], the Stochastic Segment Model of Ostendorf et al. [50, 52], and the Dynamical System Segment Model of Digalakis et al. [18–21]. All of these, like SUMMIT, model entire phonetic units. They differ in the way segments are proposed and modeled.

Figure 3-1 shows a block diagram of the SUMMIT speech recognizer coupled to TINA, a natural language (NL) processing system [65]. We will briefly describe each of the components in the next few sections. In this thesis, we did not use any of the components below the dashed line.

Briefly, the speech waveform is digitized and fed into the signal processing component, where frame-based measurements are computed. These frame-based measurements are examined to form an initial segmentation of the utterance. This initial segmentation consists of a network of interconnected segments. Paths through this segment network represent different ways of dividing, or segmenting, the utterance into phonetic units. Segment-based acoustic measurements are computed for each segment in the network. The lexical access search determines the optimal transcription of the utterance by jointly optimizing over all segmentations and classifications of the associated segments. This process involves acoustic modeling, lexical (or phonological or pronunciation) modeling, and crude language modeling. The output of the lexical access search is either the first-choice transcription, a list of the N-best transcriptions, or a word graph containing the N-best transcriptions in a more compact representation. The output of the lexical access search can be re-ordered using higher-order n-gram language models. We discuss each of these components briefly in the next few sections, with the exception of word graphs which are presented in detail in Chapter 4.

The remaining components, which were not used in this thesis, operate as follows. The N-best hypotheses resulting from the n-gram language modeling component can be further re-sorted based on more accurate context-dependent acoustic modeling. Finally, any of the N-best lists can be input into the TINA system for natural language

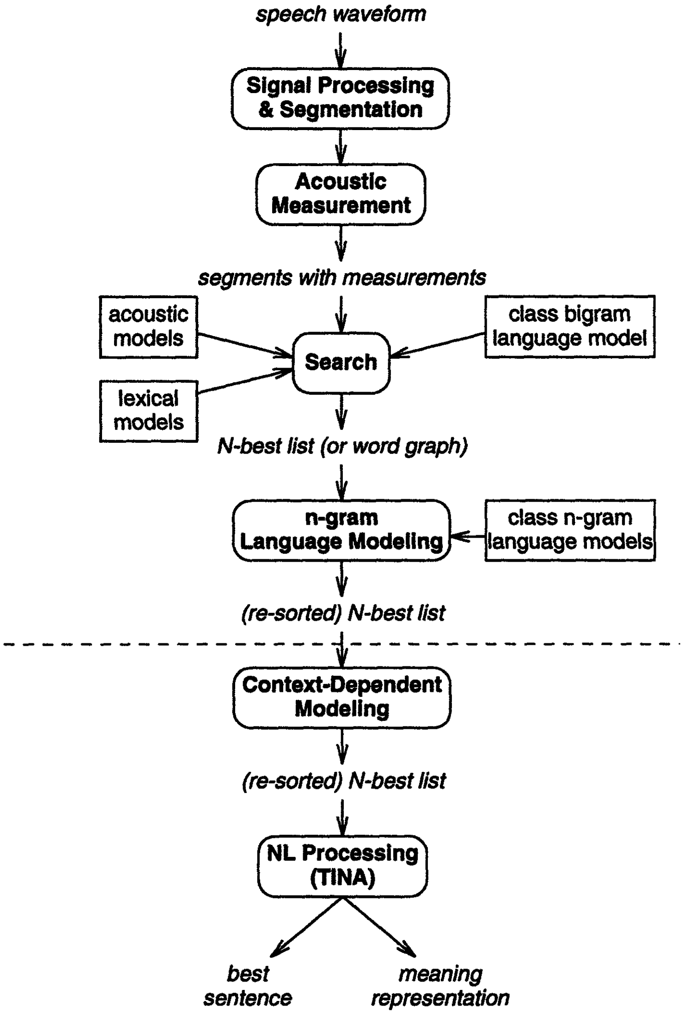


Figure 3-1: Block diagram of the SUMMIT/TINA spoken language system.

processing [65]. TINA can be used to understand utterances, or it can be used to filter N-best lists with its powerful language modeling capabilities.

3.1 Signal Processing

The signal processing used by the SUMMIT system involves transforming a 16 kHz, 16-bit sampled waveform to 14 mel-frequency cepstral coefficients (MFCCs). These MFCCs are computed for fixed-rate frames every 5 ms. The segmental nature of SUMMIT surfaces after the initial signal processing and is discussed in Section 3.2.

To derive the MFCCs, a 256-point discrete Fourier transform (DFT) is first computed for every frame from the pre-emphasized waveform using a 25.6 ms Hamming window. These spectral coefficients are passed through a set of 40 triangular filters along the mel-frequency scale, resulting in mel-frequency spectral coefficients (MFSCs) [49, 60].[2] Finally, the MFSCs are transformed from the spectral domain to the cepstral domain by taking logarithms and applying the inverse discrete Fourier transform (IDFT).

MFCCs are a popular signal representation for several reasons. They are quickly and easily computed compared to more elaborate (and realistic) auditory models, while approximating the non-linear frequency scale of the human auditory system. Compared to spectral coefficients (e.g., MFSCs), cepstral coefficients tend to be more statistically independent with respect to one another. This means that simpler probabilistic models (e.g., diagonal as opposed to full-covariance Gaussian models) can be employed in modeling them.

3.2 Initial Segmentation

SUMMIT is segmental in that it creates a *segment network* within which acoustic modeling is performed. This network of segments is created by first locating possible acoustic landmarks. In SUMMIT, the locations of these possible acoustic landmarks, or acoustic boundaries, are hypothesized in a bottom-up fashion starting from the frame-based

[2] The mel-frequency scale is linear below 1 kHz and logarithmic above 1 kHz. It is an approximation of the frequency scale of the human auditory system.

MFCCs. Where there are abrupt changes along the time axis, acoustic boundaries are proposed. Associated with each boundary is a score representing the confidence that the boundary is actually the boundary between two sub-word (phonetic) units. This boundary scoring is local, based on the MFCCs of the adjoining frames. The result is a set of scored acoustic boundaries.

Acoustic segments (arcs) are formed by connecting acoustic boundaries (nodes) to form a segment network or graph. These connections are formed by proposing possible segments that can span several boundaries. As a result, segments generally overlap each other. The goal of this process is to propose acoustic segments that include exactly one phonetic unit. Ideally, the segment network contains exactly one segment for every phonetic unit in an utterance. However, since this bottom-up process of proposing segments has no higher-level information about the identity of phonetic units, it must over-generate segments to help insure that actual segments are not missed. Part of the difficulty is due to the fact that phonetic boundaries vary in distinctiveness due to co-articulation. Eventually, during the lexical access search, the segment network is traversed to identify the relevant choice of segments.

3.3 Segmental Acoustic Measurements

Once the segment network has been constructed, SUMMIT computes acoustic measurements for each segment. The real power of a segmental system is that it can make acoustic measurements that are relevant to *entire* phonetic units. In the case of SUMMIT, these measurements include duration, MFCC averages, and a set of automatically learned acoustic measurements [57]. The set of learned measurements includes time averages of parameters over different parts of a segment, average spectral peak frequencies, and average change of spectral peaks. These last two types are related to formant frequencies and their slopes. Many of these measurements can be made at or beyond the boundaries of a particular segment. These segment-external measurements are useful in capturing co-articulation effects in adjoining segments. For example, formant transitions in neighboring vowels can be a powerful clue when determining the identity of consonants. Altogether, 36 acoustic measurements are made plus duration.

3.4 Acoustic Models

Once the acoustic measurements have been computed for all of the segments in the segment network, the segments are classified phonetically. SUMMIT does not make hard decisions for each segment at this point, but instead scores each segment probabilistically against each phonetic sub-word unit. These scores are stored for future reference in the lexical-access search described in Section 3.7.

The probabilistic score for each segment is essentially the conditional probability $P(\vec{a}_s \mid p)$, where $\vec{a}_s$ is the vector of acoustic measurements for segment s, and p is a particular phonetic unit. Principal component analysis is used to reduce the correlation of the acoustic measurements, transforming $\vec{a}_s$ to $\vec{a}'_s$.

SUMMIT approximates the conditional probability $P(\vec{a}_s|p)$ using mixture Gaussian models:

$$P(\vec{a}_s \mid p) \approx \sum_{i=1}^{M_p} w_{p,i} P(\vec{a}'_s \mid \vec{\mu}_{p,i}, \vec{\sigma}^2_{p,i}).$$

Here, M_p is the number of mixtures for phonetic unit p; $w_{p,i}$ is the mixture weight for mixture i; and $\vec{\mu}_{p,i}$ and $\vec{\sigma}^2_{p,i}$ are the mean and variance parameters, respectively, for the multi-dimensional diagonal Gaussian density.

In SUMMIT, the modeling for segment duration is separated from the modeling for the other acoustic measurements. The actual number of mixtures utilized by each phonetic unit is dependent on the amount of available training data, but the maximum we allowed for each phonetic unit in our experiments were 16 mixtures for duration and 64 mixtures for the other acoustic measurements.

3.5 Lexical Models

The lexical models represent the pronunciation of words in the vocabulary. In SUMMIT, these lexical models are more sophisticated than strings of sub-word phonetic units: they are phonetic networks, or graphs [75]. SUMMIT uses phonetic networks in order

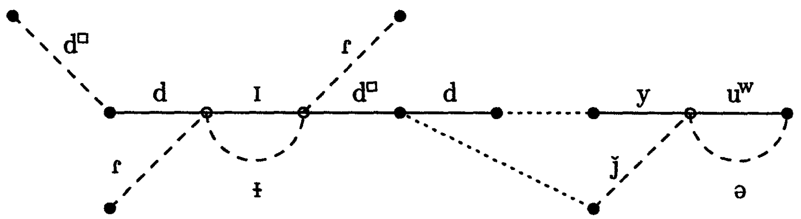


Figure 3-2: Example of pronunciation network connections. Connected pronunciation networks for the two words "did you." Solid arcs represent base-form pronunciations, dashed arcs are the result of applying the phonological rules, and dotted arcs indicate inter-word connections. Solid nodes indicate word begin/end nodes where inter-word connections are possible, and hollow nodes are word-internal nodes. In this example, the [ɾ] alternative at the top in "did" cannot attach to "you," but may attach to other words (starting with a vowel).

to model alternative pronunciations compactly.

The alternative pronunciations are not all designed by hand. The majority of them are the result of applying phonological rules to base-form phonemic pronunciations found in an on-line dictionary. The phonological rules were designed by hand, and are capable of deleting and/or adding phone arcs to pronunciation networks. These rules are not only applied within words but also *between* words.

The application of rules between words makes inter-word connections relatively complicated as can be seen in Figure 3-2. In this example, we show pronunciation networks for the words "did you" with their corresponding connections. The figure shows the expansion due to the phonological rules, including those that cross word boundaries. In this example, [dɪd$^{\square}$dyu^{w}] is the base-form pronunciation, and [dɨd$^{\square}$ǰə] is an alternative pronunciation resulting from the application of intra-word and inter-word rules. As the figure indicates, not all begin/end nodes can connect to all others. The inter-word phonological rules dictate which connections are sensible.

The arcs in the lexical models, the lexical arcs, have scores or weights associated with them. These weights are trained using a corrective training algorithm [75] and are designed to favor pronunciations that help recognition performance. The weights are needed because the phonological rules tend to over-generate arcs, and this over-generation can increase word confusions.

3.6 Class n-Gram Language Models

SUMMIT makes use of n-gram language models in the lexical access search and in N-best re-sorting. In the lexical access search (Section 3.7) a class bigram is used to constrain word sequences. The reason for a bigram language model is for computational efficiency. Because the first stage of the search is a dynamic programming search similar to Viterbi decoding [70], using more complex models is difficult because of longer language context. In a later recognition stage, a class n-gram model is used to re-sort N-best lists of complete-utterance hypotheses. In this re-sorting, SUMMIT typically uses a class 4-gram language model.

By *class*, we mean word class (e.g., names of cities and airports or days of the week). Classes are used because the quantity of training data is insufficient to properly estimate all the parameters of the n-gram models, particularly for $n \geq 3$. The classes allow words to be collapsed when estimating conditional probabilities.

The class n-gram probability $P_n(\cdot)$ for word w_j is computed as

$$P_n(w_j) = P(w_j \mid c(w_j)) \cdot P(c(w_j) \mid c(w_{j-1}), \ldots, c(w_{j-n+1})),$$

where $c(w_j)$ is the word class for word w_j. This equation shows the n-gram approximation, where only the preceding $n-1$ are included in the condition, and the collapsing of words into classes. Not only are the conditioning words collapsed into classes, but prediction is also performed using a class and a class-dependent unigram. Compared to class-conditional modeling, $P(w_j \mid c(w_{j-1}), \ldots, c(w_{j-n+1}))$, and straight word modeling, $P(w_j \mid w_{j-1}, \ldots, w_{j-n+1})$, this class-prediction modeling is more robust when training data is sparse. Note that not all words belong to classes, so many of the probabilities still depend on particular words.

However, some smoothing of the n-gram probabilities $P_n(\cdot)$ is necessary because some words were too rare in the training data to yield reliable estimates for them. One form of smoothing involves interpolating with lower-order n-gram models. The interpolated probability $P_n^i(\cdot)$ is computed as follows:

$$P_n^i(w_j) = \lambda_n P_n(w_j) + \cdots + \lambda_1 P_1(w_j).$$

The λ's are a function of word-condition counts observed in training data and favor the higher-order models if the training data are sufficient. This interpolated n-gram model is not unique to SUMMIT. Jelinek [33] presents a good tutorial of the issues related to n-gram models.

Another type of smoothing affects $P_1(w_j)$. A "floor" constant γ is added to all unigram counts. Thus,

$$P_1(w_j) = \frac{n(w_j) + \gamma}{\sum_w [n(w) + \gamma]},$$

where $n(w)$ is the unigram count for w. Typically, $\gamma = 20$.

Finally, one additional type of smoothing is employed to deal with words that were not well represented in training yet occur in testing. Because training conditions do not always match testing conditions, the interpolated n-gram models are smoothed with a uniform unigram model as follows:

$$P(w_j \mid w_{j-1}, \ldots, w_0) \approx \delta/||\mathcal{V}|| + (1 - \delta) \cdot P_n^i(w_j),$$

where $||\mathcal{V}||$ is the size of the vocabulary $\mathcal{V}$. The effect of this δ smoothing is to create a "floor" on the probabilities of words: the probability of any word is never less than $\delta/||\mathcal{V}||$ no matter its context. Empirically, δ was set to 0.02 to optimize recognition performance within the ATIS domain.

3.7 Lexical Access Search

In SUMMIT, the lexical access search is where the speech signal is decoded. Here the acoustic model scores, the lexical constraints, and the language model scores are combined to find the most likely word sequence(s). In general, the number of possible word sequences is extremely large, of order $O(\ell^v)$, where v is the vocabulary size and ℓ is the sequence length. However, through the use of pruning and clever search algorithms, this entire search space can be effectively traversed in reasonable time. SUMMIT uses a two-pass approach to lexical access. The first pass, forward in time, is a dynamic programming Viterbi search that finds the single-best scoring word sequence. If the N-best word sequences are desired, a second pass, backward in time, is used. This second pass is an A* search, or possibly an A* word graph search as presented in Chapter 4.

3.7.1 Viterbi Forward Search

SUMMIT uses a modified Viterbi search [70], forward in time, to compute the single-best word sequence that covers an entire utterance. This dynamic programming search computes the best (partial) word sequence and its score from the beginning of the utterance to every lexical node–boundary pair. Because only the best path to every node-boundary pair is extended, the search is considerably more efficient than a direct, exhaustive search.

In the absence of any pruning, this first stage search is admissible, meaning that it will find *the* best word sequence and its score. However, in SUMMIT, and in many other systems, pruning in the form of a beam search is used. It is called a beam search because at every acoustic boundary, the number of active nodes (nodes that are allowed to be extended) is limited. In SUMMIT, this beam is so wide that this pruning introduces negligible search errors.

3.7.2 A* Backward Search

SUMMIT does not compute the N-best word sequences in the first-stage Viterbi search, even though it is possible [13, 62, 64, 73], because an A* search is more efficient in terms of both time and memory. SUMMIT's A* search is performed backwards in time from the end of the utterance and uses information computed during the forward Viterbi search [73].

Briefly, the A* search is a best-first search that uses a heuristic evaluation function [47]. This evaluation function takes into account the actual score for a (partial) sequence and a heuristic estimate for the best completion of the sequence. If this heuristic estimate is an upper bound[3] then the search is admissible. The search proceeds by enqueueing word sequences, or paths, in a priority queue, dequeueing the best, and enqueueing its extensions. This process continues until a complete path is dequeued, at which point the best complete path has been found. This process can be continued to find the N best complete word sequences.

The heuristic used by the SUMMIT system makes use of the intermediate results from the forward Viterbi search. Because the two searches are in opposite directions, when the A* search needs an estimate for the best completion of a path (to the beginning of the utterance), it can simply look up the Viterbi score. This means that if the forward and backward passes use the same models and constraints, the Viterbi-based heuristic function that SUMMIT uses is exact. In general, the A* algorithm is sensitive to the tightness of the upper bound estimate. Since SUMMIT's heuristic is exact, the A* search is as efficient as possible. SUMMIT's use of the forward Viterbi scores is similar to the tree-trellis search of Soong and Huang [67].

In Chapter 4 we describe the A* search in more detail while introducing the A* word graph search (that is now a part of the SUMMIT system). This latter search produces a graph of words that represents very long N-best lists compactly.

When performing the backward A* search, SUMMIT could use more powerful constraints such as higher-order n-gram language models. However, we have found it to be more efficient to compute a word graph and then search through it using another A* search. This search through the word graph utilizes higher-order n-gram language models, and is discussed further in Section 4.4.1.

3.8 Recognizer Output

As we have alluded to, SUMMIT can produce three forms of output for each utterance: the first choice (1-best), an N-best list, and a word graph. If all that is required is the first-choice utterance with bigram language constraints, then the A* search is not needed at all; the first-stage Viterbi search yields the desired answer. However, depending on

[3] The heuristic must be an upper bound when maximizing scores or a lower bound when minimizing scores.

how the recognizer output is to be used, the first choice word sequence may not be enough.

In the past several years, researchers have found N-best lists useful when integrating a speech recognizer to a natural language processing system. In the case of SUMMIT, N-best lists have been, and continue to be, used to link the recognizer with the TINA natural language system. The reason N-best lists are useful in this application is that the natural language system may not be able to understand the single-best recognizer hypothesis. For example, the recognizer's best hypothesis may contain an error that renders it unparseable. If the second best hypothesis did not contain the error, the natural language system could skip over the first choice to the second choice and proceed.

In general, N-best lists are useful whenever the recognizer's output is to undergo further processing. For example, as shown in Figure 3-1, the SUMMIT system applies more computationally expensive context-dependent acoustic models after an N-best list is generated. Such a list represents a drastically reduced search space in which to evaluate these more expensive models. The general practice of re-scoring and re-sorting hypotheses is called N-best re-sorting and is used by many systems [13, 51, 62–64]. In summary, N-best lists are useful for two purposes: to provide alternative recognizer hypotheses for natural language processing, and to provide a multi-stage mechanism for applying more computationally expensive modeling (e.g., context-dependent acoustic modeling) and constraints.

The problem with an N-best list is that variability in even a few parts of an utterance can swamp the lists due to the need to enumerate all combinations of word hypotheses in the regions of high variability. The list representation does not capture this variability in an efficient manner. A graph representation can represent variability in various parts of an utterance much more compactly. In Chapter 4, we discuss this problem in more detail and present an algorithm for efficiently computing such word graphs. In our study of new words in the context of the SUMMIT recognizer, we make extensive use of word graphs to capture the recognizer's uncertainty in the vicinity of a new word.

Chapter 4

Word Graphs

In this chapter we present both a word graph representation for speech recognizer output and an efficient algorithm for computing word graphs. At first glance, the topic of word graphs may not seem to be related to the new-word problem. However, the ability of the word graph representation to handle variability efficiently throughout an utterance, in terms of computation time and representation size, makes word graphs valuable in studying the new-word problem. As a bonus, we find that word graphs are convenient for recognition in general even in the absence of new words. In addition, we introduce exploratory data analysis tools based on word graphs that are helpful in examining recognizer behavior in the vicinity of new words.

4.1 Motivation

Until recently, most speech recognition systems have only been faced with the task of producing the single-best word string for a given input utterance. As a result, researchers have employed efficient algorithms, such as the Viterbi dynamic-programming search algorithm [70], to find the top-scoring word string. However, the Viterbi search has two problems: it is not easy to generate "near misses" to the top-scoring answer (e.g., N-best lists) and it depends on local constraints for its computational efficiency (i.e., it is not possible to use long-distance constraints such as those possible with natural language modeling). With recent research effort in developing speech understanding systems [78] it has become desirable either to integrate more complex language models into the search, or to have the speech recognition component provide multiple sentence hypotheses, which can then be filtered by the natural language component.

Initial work in combining speech recognition and natural language technology used a modification of the Viterbi search to provide the N-best sentence hypotheses, as proposed by Chow and Schwartz at BBN [13], and showed that at least for some tasks, the correct answer was very often in the top N sentence hypotheses for fairly small N, and therefore an N-best list would provide a useful interface between a speech recognition system and a natural language parser. Based on this success, other more efficient N-best search strategies were developed, including other modifications of the Viterbi search at BBN [62] and MIT [73] as well as algorithms based on the A* search, such as the work of Kenny et al. [35], Soong and Huang [67], and Zue et al. [73].

N-best algorithms have found widespread use in systems that combine speech recognition and natural language understanding, such as the systems at BBN [62] and MIT [73]. Although there have been efforts toward integrating the natural language constraints into the search itself, such as at MIT [26] and SRI [45], N-best strategies have remained popular not only because of their ease of implementation, but also because they greatly improve the efficiency of the development effort, since one can precompute N-best lists for a large corpus to use as input for natural language experiments.

In addition, an unanticipated but important application of N-best searches has been in speeding up the development and improvement of recognition algorithms. One can use N-best re-sorting experiments as a mechanism for applying computationally expensive constraints in order to improve recognition systems. For example, one can test a new acoustic model by using it to re-sort N-best lists rather than integrating this new model into the search directly [51,56,63]. Re-sorting N-best lists can require many orders of magnitude less computation than performing the complete search and may even allow the use of constraints that would not be possible in the complete search (e.g., acoustic models that depend on long-distance contextual factors). In the SUMMIT system, N-best re-sorting is used for class 4-gram language modeling and for context-dependent acoustic modeling that includes inter-word context dependency (see Figure 3-1).

While N-best search strategies have been very useful, they are beginning to encounter problems as we move towards more difficult speech understanding tasks. As both the utterance length and vocabulary size grow, increasingly larger lists of sentence hypotheses are required to capture the necessary amount of ambiguity. This is due to the fact that sentence hypotheses on N-best lists often differ minimally in highly localized regions where the acoustic signal is not very robust. A graph representation, on the other hand, can capture the same information in a much more compact form, thus solving the problem.

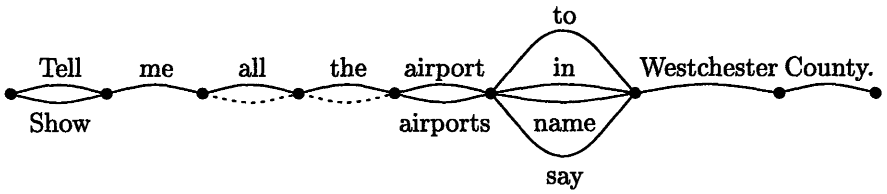
Figure 4-1(a) shows an actual N-best list computed by SUMMIT with bigram language model constraints in the ATIS domain. Examination of the N-best list reveals that the variation is somewhat localized, presumably due to a non-robust speech signal or inadequate acoustic modeling within the recognizer. Enumeration of all *combinations* of possible hypotheses from each localized region of variability quickly fills up the N-best list even though within-region variability may be relatively limited. These localized differences can be captured efficiently with a *word graph* as schematically shown in Figure 4-1(b). This figure shows that parts of the utterance, such as "me" and "Westchester County," were unchanged for $N \leq 15$ and can be shared within the word graph representation.

The difficulties with N-best lists are not restricted to the representation itself. The same problems manifest themselves in the computation of the lists. Typically, in computing an N-best list using an A* search, hypotheses are generated in a tree structure as shown in Figure 4-2. As a first-order approximation, the amount of computation is proportional to the number of branches or edges. While a tree has fewer edges than a corresponding N-best list, it has many more than a graph. Variability in the hypotheses near the beginning of the search (root of the tree) results in duplicate word hypotheses later in the search.

The problems with N-best lists are exacerbated by new, out-of-vocabulary words. If new words are present in an utterance (and there are no means for modeling the acoustics of new words), the recognizer's acoustic modeling is guaranteed to be inadequate. Typically, the recognizer hypothesizes combinations of a large number of in-vocabulary

	Tell	me	all	the	airports	in	Westchester	County.
1.	Tell	me	all	the	airport	to	Westchester	County.
2.	Tell	me	all		airports	in	Westchester	County.
⇒ 3.	Tell	me	all	the	airports	in	Westchester	County. ⇐
4.	Tell	me		the	airport	to	Westchester	County.
5.	Tell	me	all	the	airport	name	Westchester	County.
6.	Tell	me		the	airports	in	Westchester	County.
7.	Show	me	all	the	airport	to	Westchester	County.
8.	Show	me	all		airports	in	Westchester	County.
9.	Tell	me	all	the	airport	say	Westchester	County.
10.	Tell	me	all	the	airport	in	Westchester	County.
11.	Show	me	all	the	airports	in	Westchester	County.
12.	Tell	me		the	airport	name	Westchester	County.
13.	Tell	me		the	airport	say	Westchester	County.
14.	Tell	me		the	airport	in	Westchester	County.
15.	Show	me	all	the	airport	name	Westchester	County.

(a) N-best list



(b) Graph

Figure 4-1: Comparison of N-best list and graph representation. In (a), the correct string is at the top of the list, and corresponds to the third hypothesis. In the graph representation (b), dotted arcs are null transitions that allow words to be skipped.

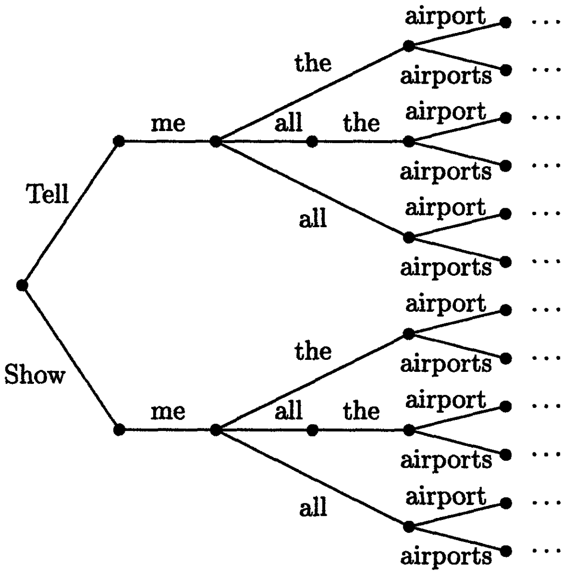


Figure 4-2: Search tree expansion. This figure schematically shows part of the search tree corresponding to the N-best list of Figure 4-1(a). It shows that variability early in the search (i.e., near the root of the tree) results in duplicate expansions that are not shared.

	I'd like to book a round-trip flight from	*Kansas*	City to	*Chicago.*
1.	I'd like to book a round-trip flight from	Denver	city to	travel.
2.	I'd like to book a round-trip flight from	and the	city to	travel.
3.	I'd like to book a round-trip flight from	Denver	city to	stop.
4.	I'd like to book a round-trip flight from	tens the	city to	travel.
5.	I'd like to book a round-trip flight from	ten's the	city to	travel.
6.	I'd like to book a round-trip flight from	and the	city to	stop.
7.	I'd like to book a round-trip flight from	Denver	city to	fly no.
8.	I'd like to book a round-trip flight from	tens the	city to	stop.
9.	I'd like to book a round-trip flight from	ten's the	city to	stop.
10.	I'd like to book a round-trip flight from	Denver	city to	Atlanta.
11.	I'd like to book a round-trip flight from	and the	city to	fly no.
12.	I'd like to book a round-trip flight from	and the	city to	Atlanta.
13.	I'd like to book a round-trip flight from	tens the	city to	fly no.
14.	I'd like to book a round-trip flight from	ten's the	city to	fly no.
15.	I'd like to book a round-trip flight from	Denver	city to	five.

Figure 4-3: N-best list in the presence of new words. The correct string is at the top of the table and contains two out-of-vocabulary words: "Kansas" and "Chicago." In this example, *all* variability for $N \leq 15$ is associated with the new words.

words for regions containing a new word in an attempt to account for its acoustics. Such a localized explosion in word hypotheses is very problematic for N-best lists as demonstrated in Figure 4-3. This example, also from the ATIS domain, contains two words not in the system vocabulary: "Kansas" and "Chicago." In this example, the only variability for $N \leq 15$ is in the regions of the new words; the hypothesized words for the rest of the utterance remain unchanged (and correct). A graph representation could capture this localized variability more efficiently. In terms of computation, the graph representation allows divergent paths due to ambiguity to be merged. This merging results in considerable computational time savings.

4.2 A* Word Graph Search Algorithm

Our A* word graph search algorithm[1] is based on the A* N-best algorithm used by the SUMMIT system. We first describe the general A* algorithm, then the N-best algorithm, and finally the word graph algorithm.

4.2.1 A* Search

An A* search [8,47] is a best-first search with a particular evaluation function $f^*(p)$ for a (partial) path, or hypothesis, p in the search space:

$$f^*(p) = g(p) + h^*(p).$$

Here, $f^*(p)$ is the estimated score of the best complete path containing p, $g(p)$ is the actual score for p from the beginning of the search, and $h^*(p)$ is a heuristic estimate of the best-scoring completion of p. The search makes use of a priority queue[2] which ranks entries using the scoring function f^*. In general, the A* search falls somewhere between a best-first and a breadth-first search, depending on the quality of the heuristic and the actual (data-dependent) search space.

[1] Previously, in [27] we called this algorithm the A* word network search algorithm.

[2] The priority queue is often called a sorted stack in speech recognition literature after its use in the stack-decoding algorithm [32], which is closely related to the A* algorithm.

The A* search begins with one entry in the queue, an empty path. The search is iterative and proceeds as follows at each iteration: the top-scoring entry in the queue is dequeued (removed from the queue), it is extended by one unit (e.g., a single word, syllable, or phone) in all possible ways, and each of these extensions are enqueued (inserted in the queue). The search terminates when the first complete path (e.g., that spans an entire utterance) is dequeued. The search itself is *admissible,* meaning it is guaranteed to find the best-scoring complete path if the scoring function h^* has the following two properties:[3]

- *admissibility:* $h^*(p) \geq h(p)$ meaning that the estimated best-completion score $h^*(p)$ is an upper bound on the actual best-completion score $h(p)$; and
- *monotonicity:* $h^*(p') + s \leq h^*(p)$, where p' is an extension of path p, and s is the actual score for the extension between p and p'.

When a path p is complete, its score $f^*(p) = f(p)$ is no longer an estimate since $h^*(p) = h(p) = 0$. Because all other partial paths in the queue have upper bounds on their scores that are less than $f(p)$, the completed path p must be the highest-scoring path. The search will find the best-scoring path and is therefore admissible.

4.2.2 A* N-Best Search in SUMMIT

To efficiently apply the A* search in spoken language systems, it is important to have as tight a bound as possible for $h^*(p)$, since the number of path extensions needed to find the best-scoring path decreases as this estimate approaches the actual score h for the completion of the partial path. We can use a Viterbi search to compute this upper bound by searching for the best completion score. In SUMMIT, a two-stage search strategy is used. The first stage is a Viterbi search that computes the best score from the beginning of an utterance to every lexical node–time pair (ℓ, t).[4] The second stage is an A* search, but it is *backward* in time. Therefore, g is the actual score computed so far by the A* search from the end of the utterance to (ℓ, t). Because the first-stage search

[3] The conditions are formulated for maximizing additive scores.

[4] Lexical nodes are nodes in the word pronunciation networks (see Figure 4-7).

was in the opposite direction, we can use the Viterbi scores for h^*, the upper-bound estimate for the best completion of p to the beginning of the utterance. In SUMMIT, the A* search operates at the *word* level, extending partial paths a word at a time. The same techniques can be applied at other levels (e.g., syllables or phones).

Of course, if all that is desired is the single-best word string hypothesis, there is no need for the second-stage A* search; the first-stage Viterbi search yields the answer. However, if an N-best list is desired, we can run the A* search until N complete hypotheses are found. If the modeling and constraints are identical in the Viterbi and A* searches then the estimate h^* is exact: $h^* = h$. This is the case with the SUMMIT system. This exact heuristic score h^* results in a very efficient A* search. In finding the best-scoring path, the search only expands partial paths that are part of the best-scoring path (or on rare occasions, paths that have exactly the same best score). Another consequence of the first-stage Viterbi search is that it computes the best score for the whole utterance. During the A* search, this score can be used to set a relative score threshold θ, which can be used to prune path extensions.

Figure 4-4 shows the A* algorithm used by SUMMIT to compute N-best lists. It begins by putting an empty path into the queue. Next, the *string_reached*[ ] lookup table (e.g., hash table) is initialized to false for all times and word strings that are possible. This lookup table is used to implement the pruning based on word strings. Within each iteration, the best path, called the current path, is dequeued and checked to see if it is complete (i.e., spans the entire utterance). If it is complete, the word string for the current path is added to the growing N-best list. Otherwise, all possible single-word extensions of the current path whose score f^* is above the relative score threshold θ are determined. Each new path formed by connecting each word extension (edge) to the current path is checked to see if it is subject to word-string pruning. Its time span and word string are checked in the *string_reached*[ ] lookup table. If there is already an entry there, there is no need to put the new path in the queue; a better path with an identical word string and time span has already been found and placed in the queue. This word-string pruning eliminates paths that differ only in internal alignment, keeping only the best-scoring one. This results in the N-best distinct word strings instead of

```
Initialization
$n \leftarrow 0$
queue ← $\emptyset$
enqueue_path(queue, EMPTY_PATH)
for all time, word_string do
  string_reached[time, word_string] ← FALSE
end for
Search
while $n < N$ and not empty(queue) do
  current_path ← dequeue_best_path(queue)
  if complete(current_path) then
    output(word_string(current_path))
    $n \leftarrow n + 1$
  else
    for each word_edge ∈ word_extensions(current_path) do
      new_path ← current_path + word_edge
      if $f^*(new\_path) \geq \theta$ then
        if not string_reached[end_time(new_path), word_string(new_path)] then
          enqueue_path(queue, new_path)
          string_reached[end_time(new_path), word_string(new_path)] ← TRUE
        end if
      end if
    end for
  end if
end while
```

Figure 4-4: SUMMIT's A* algorithm with word string pruning.

the N-best alignments.

4.2.3 Algorithm for Word Graphs

If the A* algorithm utilizes local constraints (e.g., word bigram language model), then all partial paths that end at a particular lexical node–time pair (ℓ, t) will share the same path extensions. With local constraints, the different histories of these partial paths do not matter: the partial paths are indistinguishable except on the basis of score g because they end at the same time and position within the same word. If we keep track of the endpoints (ℓ, t) of partial paths then we can merge paths that share endpoints. This merging creates a *graph* instead of the tree typical of A* searches.

Our A* word graph search consists of the basic A* search described above with the addition of a path merging and pruning step. Our algorithm for computing word graphs differs from SUMMIT's N-best algorithm in four ways:

1. an edge is added to the word graph for every partial path;

2. the word-string pruning based on *string_reached*[ ] table is removed;

3. a partial path extension is enqueued if and only if it is the best-scoring partial path to reach a particular lexical node–time point (ℓ, t) so far; and

4. the search runs until the entire queue is empty instead of N complete paths being found.

The filtering of paths to be enqueued (3) is responsible for the considerable computational savings. This pruning means that paths sharing a particular endpoint (ℓ, t) are later extended *simultaneously.*

Figure 4-5 shows the A* word graph algorithm in detail. It begins with the queue containing a single empty path. Then, the *best_so_far*[ ] lookup table is cleared. This lookup table is used to keep the best path so far to all (ℓ, t) points to check for possible path pruning. The algorithm iterates until the priority queue is completely empty, since only those paths whose f^* score was above the relative score threshold θ were originally placed in the queue. The significant difference between the N-best and word graph algorithms is what happens to newly extended paths. The *best_so_far*[ ] table is consulted to see if another path, *previous_best*, has reached the (ℓ, t) point of the new path. If one has, its score is compared to the score for the new path. If the new path has the lesser score, it is not placed in the queue. In this case, the new path is pruned in the sense that it is not placed in the queue (and wil not affect subsequent search iterations), but it forms a word edge. If the new path has a better score, it replaces the previous best path in the queue and in the lookup table. The net result is that there is at most one path in the queue that ends at a particular (ℓ, t) point: the one with the best score. All paths that reach (ℓ, t) are *merged* together in building the graph and extended *simultaneously.*

Figure 4-6 illustrates a subtle issue that involves merging paths and the order of path extension in the A* search. In our A* search, paths are dequeued and extended in

```
————————————————Initialization————————————————
queue ← $\emptyset$
enqueue_path(queue, EMPTY_PATH)
for all time, lexical_node do
  best_so_far[time, lexical_node] ← $\emptyset$
end for
————————————————Search————————————————
while not empty(queue) do
  current_path ← dequeue_best_path(queue)
  if not complete(current_path) then
    for each word_edge $\in$ word_extensions(current_path) do
      new_path ← current_path + word_edge
      if $f^*$(new_path) $\geq \theta$ then
        previous_best_path ← best_so_far[end_time(new_path), end_lexical_node(new_path)]
        if previous_best_path = $\emptyset$ then
          enqueue_path(queue, new_path)
          best_so_far[end_time(new_path), end_lexical_node(new_path)] ← new_path
        else if $f^*$(new_path) > $f^*$(previous_best_path) then
          dequeue_path(queue, previous_best_path)
          enqueue_path(queue, new_path)
          best_so_far[end_time(new_path), end_lexical_node(new_path)] ← new_path
        end if
        output(word_edge)
      end if
    end for
  end if
end while
```

Figure 4-5: A* word graph algorithm.

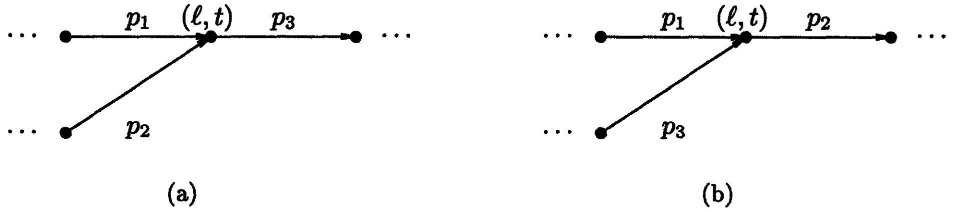


Figure 4-6: Order of path extension in word graph algorithm. In both (a) and (b), the partial paths are generated in the order p_1, p_2, p_3. In (a), path p_2 reaches the point (ℓ, t) before p_1 is extended. The better scoring of the two, according to f^*, will later be extended to form p_3 and its extensions. Thus, paths p_1 and p_2 will be extended simultaneously. In (b), path p_1 is extended to form p_2 before p_3 reaches (ℓ, t). However, it must be the case that $f^*(p_1) \geq f^*(p_3)$, otherwise p_3 would have reached (ℓ, t) before p_1 was extended. This is important because it means that in (b), the score for the extensions of p_1, including p_2, will have the correct (best) scores, and that later a path such as p_3 cannot change their scores when it later merges with one of their ancestors.

decreasing order of score f^*, but they are enqueued in no particular order. Therefore, it is possible that a relatively poor-scoring path could reach a particular point (ℓ, t) first and be placed in the queue. However, the nature of the A* search guarantees that such a path will not be extended unless it is *the* best to reach (ℓ, t). If another path later reaches (ℓ, t) with a better score, the original path will not have already been extended. The result of this order of path extension is that all paths reaching the point (ℓ, t) are extended simultaneously exactly once. Never does a path extension need to be rescored because the best score of one of its ancestors changed due to path merging. This complete elimination of duplicate path extensions is how the A* word graph algorithm gains its computational efficiency over the N-best A* algorithm.

4.2.4 Word Graph Output

The A* word graph search algorithm produces a directed acyclic graph (DAG). The nodes, or vertices, of the graph represent word-initial lexical node–time pairs (ℓ, t), and the edges, or branches, represent individual word hypotheses. Each edge e has associated with it:

- a word label,
- a score $s(e)$ consisting of acoustic and lexical scores (but not bigram scores),
- a forward score $g(e)$ containing bigram scores, and
- a backward score $h(e)$ containing bigram scores.

The forward and backward scores $g(e)$ and $h(e)$ are the scores for the best-scoring paths terminating at edge e (inclusive of e) to the beginning and end of the utterance, respectively. These scores contain bigram language model scores and can be combined with the edge score $s(e)$ to yield the score $f(e) = g(e) + h(e) - s(e)$ of the best-scoring complete-utterance path that contains the edge e. Finally, the word graph contains the set of word edges $\mathcal{E} = \{e : f(e) \geq \theta\}$. Thus, the word graph contains all word edges in an N-best list containing all alignments, where N is a function of the threshold θ.

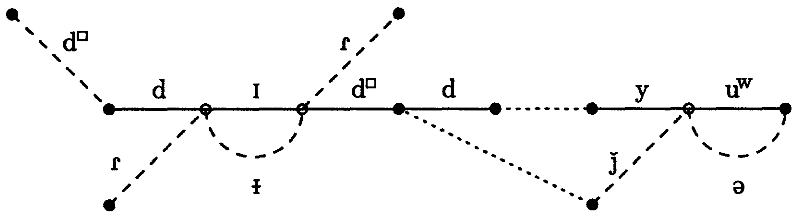


Figure 4-7: Example of pronunciation network connections. Connected pronunciation networks for the two words "did you." Solid arcs represent base-form pronunciations, dashed arcs are the result of applying the phonological rules, and dotted arcs indicate inter-word connections. Solid nodes indicate word begin/end nodes where inter-word connections are possible, and hollow nodes are word-internal nodes. Partial paths (coming from the right in the backward pass) that end at different word-initial nodes of "you" cannot be merged during construction of the word graph because they have different connectivity to other words (to the left), as is the case with the word "did" (the dotted lines).

Our word graphs tend to have a large number of edges in them, because they contain all *alignments* of all N-best strings above the relative score threshold. The number of edges is further increased by the fact that pronunciation networks for words in SUMMIT typically have multiple word-initial nodes at which merging takes place. Because the connections between words are constrained by the inter-word rules, we have found it convenient to perform partial path merging at word-initial nodes (searching backward in time). Figure 4-7 (same as Figure 3-2) shows an example of the connections between the words "did" and "you." In it, the word "you" has two word-initial nodes associated with [y] and [ǰ]. We cannot merge partial paths (coming from the right) that end at these two distinct word-initial nodes because doing so would not allow proper connection to other words to the left, such as "did." The word-final nodes must be kept distinct in the word graph so that the inter-word connectivity due to the phonological rules is respected. This results in additional word edges due to less partial path merging than if the two word-initial nodes for "you" were combined. However, we have no choice because of the way the inter-word rules interact with the pronunciation networks in the SUMMIT system.

4.3 Efficiency

The advantages of the A* word graph search include more compact representation and faster computation as compared to the A* N-best search for the same relative search depth, especially for very large N. Since the actual representation size and computational demands of word graphs and N-best lists are dependent on speech data, we performed some experiments comparing the two algorithms. Theoretically, the A* N-best search is exponential in time and space requirements in the worst case. However, in practice the search is tractable. To evaluate real-world performance we have to examine the operation of the algorithm running on real speech data. Therefore, we chose an empirical approach to measuring performance.

We compared various measures of computational requirements and representation size as a function of search depth and utterance length. We defined the search depth to be the score threshold relative to the best-scoring complete-utterance hypothesis. Both the N-best and word graph searches can produce all word strings within a specified score threshold simply by running the searches until the queues become empty (since no hypotheses that fall below the threshold are ever enqueued). One difference between the effective output of the N-best list and word graph algorithms is that the word graphs contain all alignments of all word strings above the threshold whereas the N-best lists contain just the *best* alignment of each of the distinct word strings. Therefore, the word graphs contain *more* information that might prove useful for subsequent processing. Even so, we have found significant efficiency improvements with the word graph search.

4.3.1 Experimental Conditions

The corpus used for this evaluation was a subset of the DARPA November 1992 ATIS evaluation test set [53]. To reduce the amount of computation needed, only the utterances from the first session for each speaker were used. We also discarded a few of the longest utterances, because we were not able to compute the N-best search to the search depth used in the experiments. The reason for this was that the A* N-best searches for these utterances were too computationally expensive to perform our experiments.

In contrast, we had no difficulty with the word graph searches on these same utterance. Nevertheless, we discarded the utterances from our evaluation. This left us with 196 utterances from 29 speakers, which we believe were adequate to demonstrate the computational and representational efficiency improvements of the word graph search algorithm.

The recognition system we used was the SUMMIT system, described in Chapter 3. For these experiments, we used context-independent acoustic models and a bigram language model. This stripped-down version of the system had a first choice word accuracy of 76.4% on these 196 utterances.

For these experiments we compared various measures of efficiency versus search depth, but the relevant range of search depths depends on the requirements of subsequent processing stages. If the word graphs are to be used for the initial stage of a multi-stage search, then the depth needed in the first stage depends on the relative strengths of the constraints used in the later stages of the search. If later stages of the search are capable of large scoring changes, we need a relatively loose threshold in the initial search so as to limit search errors that result from correct answers not being included in the word graphs.

To gauge the range of search depths of interest for our experiments, Figure 4-8 shows the percentage of correct sentences contained within a given relative score threshold θ. Note that this is a spontaneous speech task and does contain new, out-of-vocabulary words. Therefore, the sentence accuracy will not reach 100% no matter how deeply we search, since we made no attempt to address the problem of new words. For this experiment, only 76.6% of the sentences were fully in-vocabulary. This ceiling on sentence accuracy is displayed in the figure as a horizontal line. The vertical line shows the maximum score threshold $\theta = 800$ used in the following experiments.

4.3.2 Computational Time Efficiency

It is difficult to compare in complete detail the computational needs of these two algorithms, since the overall computational efficiency depends on the details of the various parts of the computation (e.g., the implementation of the priority queue). Instead, we

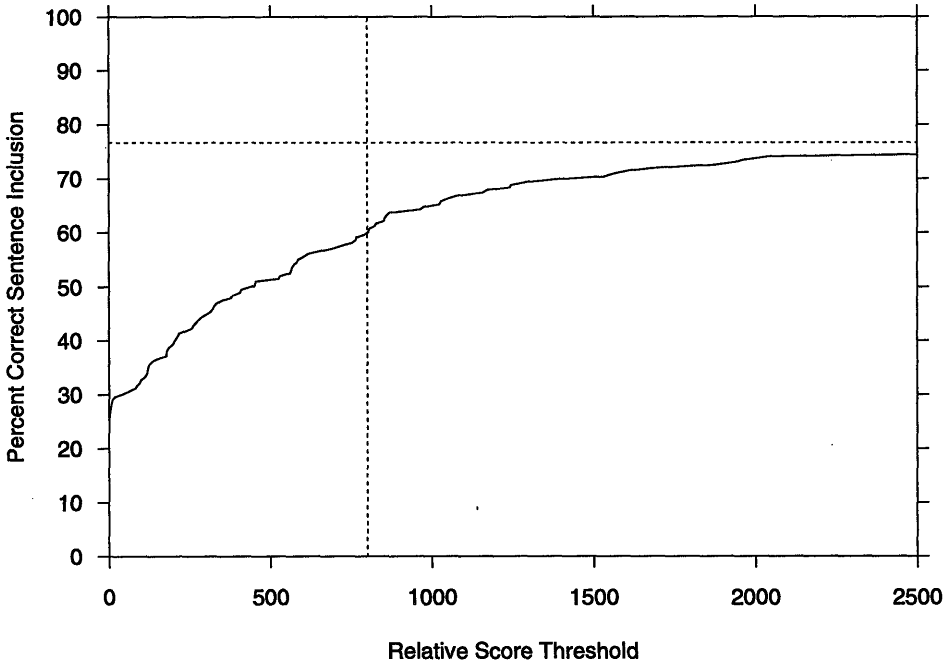


Figure 4-8: Sentence accuracy versus relative score threshold θ. These utterances contained out-of-vocabulary words; the horizontal line shows the best the system can do regardless of search depth.

have focused our attention on the computation that the two algorithms have in common: extending partial paths, or hypotheses. We used the number of partial path extensions as our measure of computation. Since extending paths is where the majority of time is spent in both algorithms, this is a reasonable measure of computation time.

Figure 4-9 displays the geometric mean across utterances of the number of path extensions needed to search to a given search depth. We chose the geometric mean because the distribution of number of path extensions is roughly log-normal. The arithmetic mean is dominated by worst-case utterances, and there is a very large variation between different utterances. This figure shows that for a given search depth, the word graph search requires fewer partial path extensions, and thus runs faster. Further, as the search depth increases, the difference between the A* search and the word graph search increases. This difference is due to the significant amount of path merging in the word graph.[5] At the score threshold $\theta = 800$, which we commonly use in recognition

[5] Recall that our N-best search also performs some limited path merging/pruning based on word

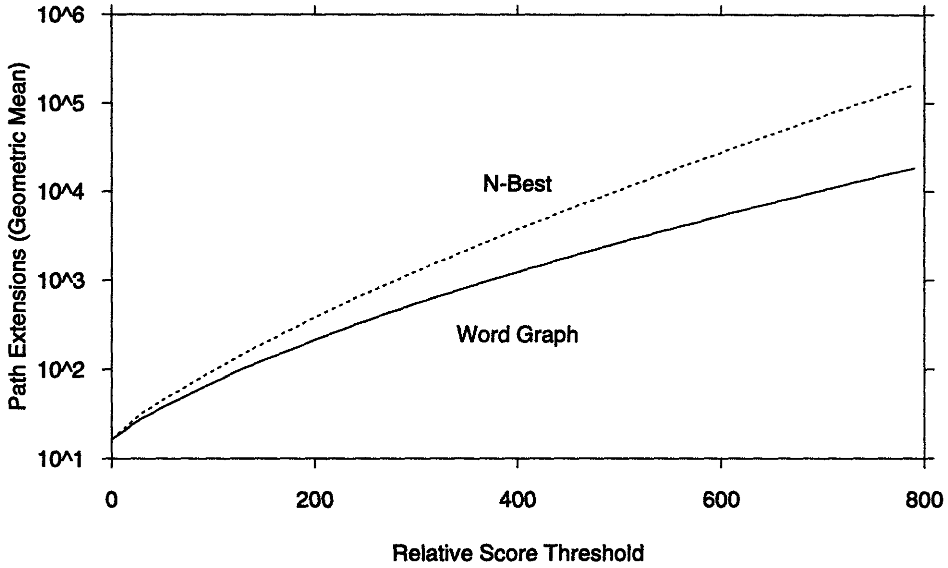


Figure 4-9: Number of partial path extensions versus relative score threshold. This figure compares the computational efficiency of the A* N-best and word graph searches for various search depths as measured by the number of partial path extensions.

experiments, the word graph computation is a full order of magnitude faster.

We have noticed a very large utterance-to-utterance variation in the search time for a given search depth. For example, the number of path extensions ranges from 562 to well over 8,000,000 for the maximum depth $\theta = 800$ in Figure 4-9. While some of this variation is certainly due to the strength of the acoustic evidence, there is also a strong dependence on utterance length. Figure 4-10 shows a scatter plot of the number of partial path extensions versus utterance duration with the search depth fixed $\theta = 600$. For both searches, we have overlayed lines produced by a scatter plot smoothing function.[6] As utterance duration increases, the required search effort increases for both algorithms, but the increase is much more substantial for the N-best search. The word graph search is better behaved, requiring about two orders of

strings.

[6] The smoother is *lowess* procedure in S [9], which produces smooth, robust, locally linear fits of the scatter plot points. The line for N-best stops at 10 seconds because some of the longer utterances required too much computation to reach the search depth and were not included in the plot. However, we had no difficulty computing the corresponding word graphs. It did not make sense to plot the N-best curve beyond the point at which its data were missing. The systematic nature of the missing data would bias the N-best curve downward.

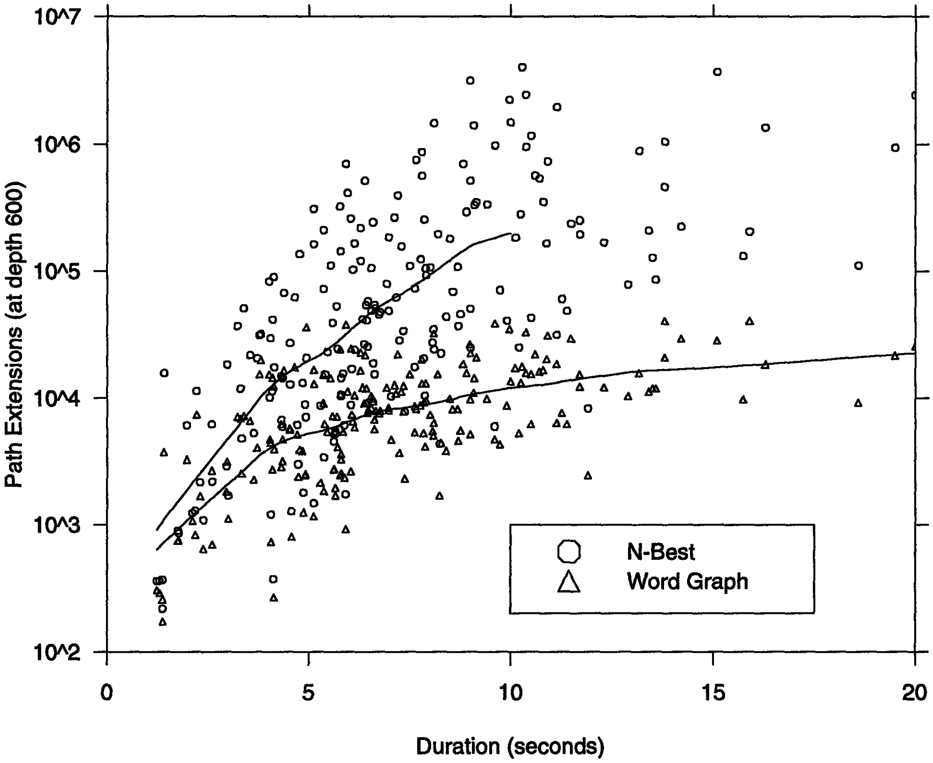


Figure 4-10: Number of partial path extensions versus utterance duration.

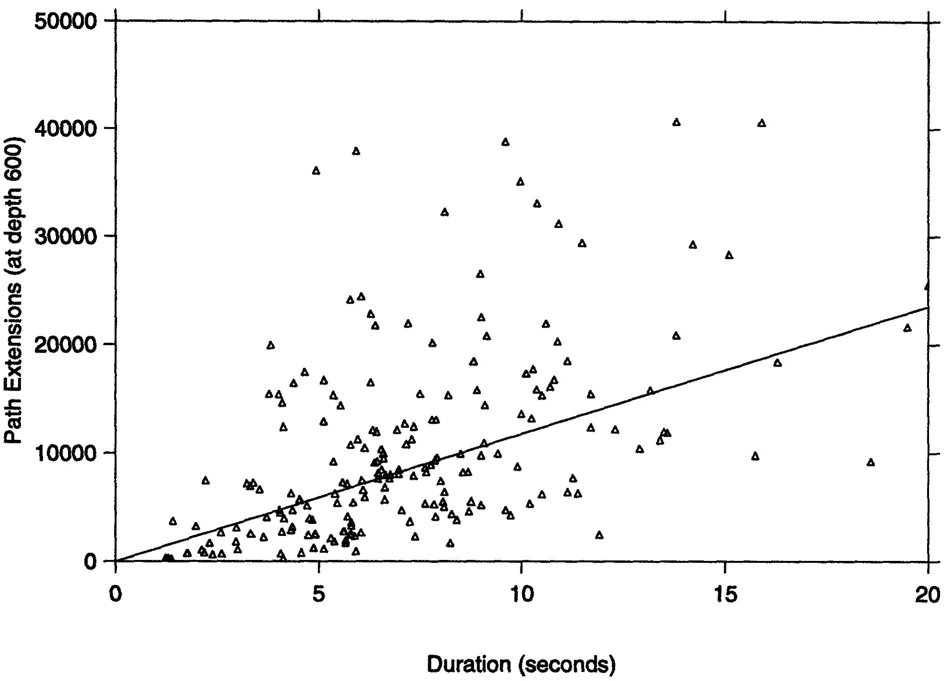


Figure 4-11: Linearity of word graph computation versus utterance duration.

magnitude fewer expansions in the worst cases.

Because of the merging that takes place during the A* word graph search, intuitively we would estimate that the amount of computation (i.e., number of path extensions) is roughly proportional to utterance duration. If the average branching factor in the graph is relatively constant we would expect the merging to yield computation proportional to duration. Figure 4-11 shows a plot of word-graph path extensions versus utterance duration with linear axes. The superimposed line is the linear fit with slope 1177edges/sec. It appears that the growth rate could be linear, but the variance is so high it is difficult to determine with certainty. The growth rate of the computation is certainly less than exponential, which we would expect for the N-best search.

4.3.3 Output Representation Size

We have examined the sizes of the N-best and word graph representations versus search depth. For N-best lists, a reasonable measure for size of representation is the number of paths, or hypotheses, removed from the queue, since every partial path dequeued

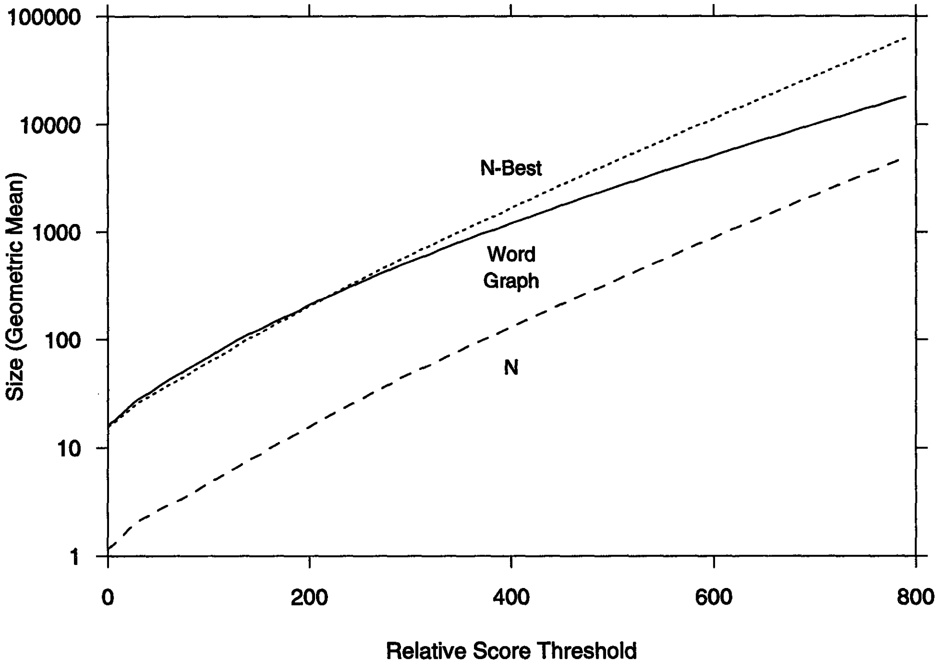


Figure 4-12: Representation size versus relative score threshold. The curve at the bottom shows the N corresponding to the N-best lists.

corresponds to a word in the N-best list (subject to word-string pruning). For word graphs, the number of edges in the graph *is* the number of partial path extensions and is the obvious choice for size of representation. Figure 4-12 displays the relative sizes of the two representations as the search depth increases. We have also plotted N, the number of distinct word strings. Again, we have plotted geometric means because the sizes approximately follow a log-normal distribution. At first glance it may appear that the word graph representation is not as efficient as the N-best list for very small N. However, we must not forget that the word graphs contain *more* information; they contain *all* possible alignments of the N-best word strings whereas the N-best lists contain only the best alignment of each of the distinct word strings. Depending on the intended use of the word graphs, it might be desirable to apply graph reduction algorithms. This would be especially useful were we concerned only with word strings (as contained in the N-best output) rather than all possible alignments of these strings. Even without such pruning, the word graphs have considerably smaller representations for all but the smallest search depths. Furthermore, the growth rate is lower, so word graphs are even more advantageous as the search depth increases.

4.3.4 Summary

In summary, we introduced the A* word graph algorithm for computing word graphs. This algorithm represents a relatively small change to the existing A* N-best algorithm used by the SUMMIT system that allows partial utterance hypotheses to merge during the search. The result of this merging is a *graph* instead of the *tree* associated with the N-best search. We showed that by avoiding opening up the search space into a tree, the word graph search algorithm can compute utterance hypotheses at least an order of magnitude faster than the N-best search for deep searches. Further, we showed that in terms of output representation size, word graphs are more compact than N-best lists despite containing significantly more information (more alignments). This additional information could be useful when post-processing word graphs, as we will see in the next section.

4.4 Post-Processing Word Graphs

Word graphs can be used in several ways. Their most obvious use is as an intermediate representation for recognition hypotheses. Like N-best lists, they can be used to interface various speech recognition and understanding components. In Section 4.4.1 we describe how our word graphs can be used in multi-stage searches. In Section 4.4.2 we demonstrate how word graphs can be useful in exploratory data analysis because they contain all individual word hypotheses, including their endpoints and scores, considered in the search process.

4.4.1 Searching Through Word Graphs

In general, a word graph is an intermediate representation useful in a multi-stage recognition/understanding search strategy. Ultimately, a single-best hypothesis needs to be selected for recognition. To do this, we can search through word graphs.

As describe in Section 4.2.4, each edge in a word graph contains a word label, an acoustic/lexical score for the edge itself, and forward and backward scores g and h. The forward and backward scores contain bigram language model scores. The presence of these scores allows easy computation of N-best lists in either the forward or backward directions. To generate an N-best list with bigram language model constraints, we can first compute a word graph to a suitable depth, and then perform an A* search through the word graph. In this A* search, no additional acoustic or lexical modeling is required; the scores in the graph are sufficient. The forward score g or the backward score h, depending on search direction, can be used as an exact heuristic for the A* search.

In fact, in performing the experiments in Section 4.3, we computed the N-best lists by searching through word graphs. We found that the direct A* N-best search was far too computationally demanding for large N, since all of the duplicate path extensions require duplicate acoustic and lexical modeling. With the word graph approach, the acoustic and lexical modeling is efficiently captured in the word graph. The A* search (with bigram language model constraints) through the word graph is very efficient since all modeling has been completed. For each word edge, the word graph contains its acoustic, lexical, *and* bigram language model scores. Therefore, the search only has to enumerate the words to produce the N-best word strings in order. This process is extremely fast because no additional acoustic or language modeling is required.

We can similarly produce N-best lists using *different* models if we wish. For example, in the SUMMIT system, higher-order class n-gram language models are often used when searching through word graphs. However, when we search through word graphs using different models we encounter the problem that the A* search is no longer guaranteed to be admissible. If the forward or backward scores contained in the word graph are used for the heuristic score h^*, a language model change could render h^* inadmissible (i.e., no longer an upper bound on h). The forward and backward scores were computed with a *bigram* language model. With a different model, say a higher-order n-gram language model, the function h can change, perhaps increasing for some partial paths. Thus, the A* search through the word graph is generally inadmissible when the models used are different from the models used in the computation of the word graph.

However, we have found that the problem of inadmissibility can be ameliorated by searching deeper than we otherwise would with an admissible search. When computing an N-best list with different models, we compute an N'-best list where $N' > N$ and re-order the list, keeping the top N hypotheses. Typically, when using class 4-gram language models, we search for N' on the order of 150 to find the 10-best list. Since searching through the word graphs is so fast, the extra N' is not a problem. This method has worked well and is used for the recognition experiments in Chapter 5.

4.4.2 Exploratory Data Analysis using Word Graphs

We have discussed how word graphs are useful for representing a recognizer's output for use in multi-stage searches. However, word graphs and the statistics derived from them can be helpful in studying recognizer behavior. Because of the way the A* word graph search algorithm builds word graphs, they contain all individual word hypotheses that would be explored in an A* N-best search down to the prescribed score threshold. In effect, the word graph represents a detailed history of the recognizer's search (N-best or word graph). For each edge, a word graph contains its word label, its acoustic/lexical score, and its forward/backward best-completion scores.

Word Lattices

One method we have found useful for displaying the contents of word graphs is what we call a *word lattice.* In a word lattice, we display words and their time spans for within an utterance. For each edge in the word graph, we display a line between its time endpoints at height s, where s is its acoustic/lexical score (the combination of the acoustic model scores and the lexical arc weights). The score s is language model–independent because it does not contain the bigram score. (However, the bigram language model score does affect the pruning that goes into the computation of the word graph.) These word lattices allow us to see the competing words across an utterance. We call them lattices to distinguish them from graphs because the word edges are not connected together in the display.

Our word graphs tend to contain a very large number of edges because they contain all alignments of all N-best strings above the relative score threshold. The fact that merging during the word graph search occurs at word-initial nodes further increases the number of word edges in the graphs. In order to simplify the display of word lattices, we have limited the number of word edges we display in two ways: by adjusting the relative score threshold and by filtering word edges with identical labels and endpoints. For all edges with identical word labels and time endpoints, we draw only the best-scoring one.

Figure 4-13 shows an example word lattice. At the top is the wide-band spectrogram, the waveform, and the time-aligned orthographic transcription. At the bottom is the word lattice, with acoustic/lexical score on the vertical axis and time on the horizontal axis. For each word hypothesis, the word lattice shows its time extent (t_1, t_2) and score s. This word lattice was plotted for relative score threshold $\theta = 300$, meaning only edges that are part of complete hypotheses that score within 300 of the best-scoring complete hypothesis are included. This represents considerable pruning compared to the typical value of $\theta = 800$ we use to compute the word graphs. This pruning was necessary to reduce the number of word hypotheses displayed. In this example, "What airlines serve Denver?" there are relatively few edges. Most of the competition between word edges is due to slight time-alignment differences. For "serve," the distinct competitors are "serving" and "serve from." For "Denver," the competitor is "dinner."

Figure 4-14 shows another word lattice. In this one, "Hi, I'm in Chicago," the word "Chicago" is out-of-vocabulary. This word lattice has considerably more edges than that of Figure 4-13 even though it is computed to the same $\theta = 300$. The reason for this is that none of the in-vocabulary words accounts for the acoustics of "Chicago" very well, and the recognizer hypothesizes many combinations of words in an attempt to account for the acoustic signal. The recognizer hypothesizes over 30 words in the place of "Chicago." However, of even more importance is that the occurrence of the new word "Chicago" causes recognizer confusion in the preceding words "I'm in," where "interested" is the highest-scoring candidate. We come back to this example later in Section 5.4.

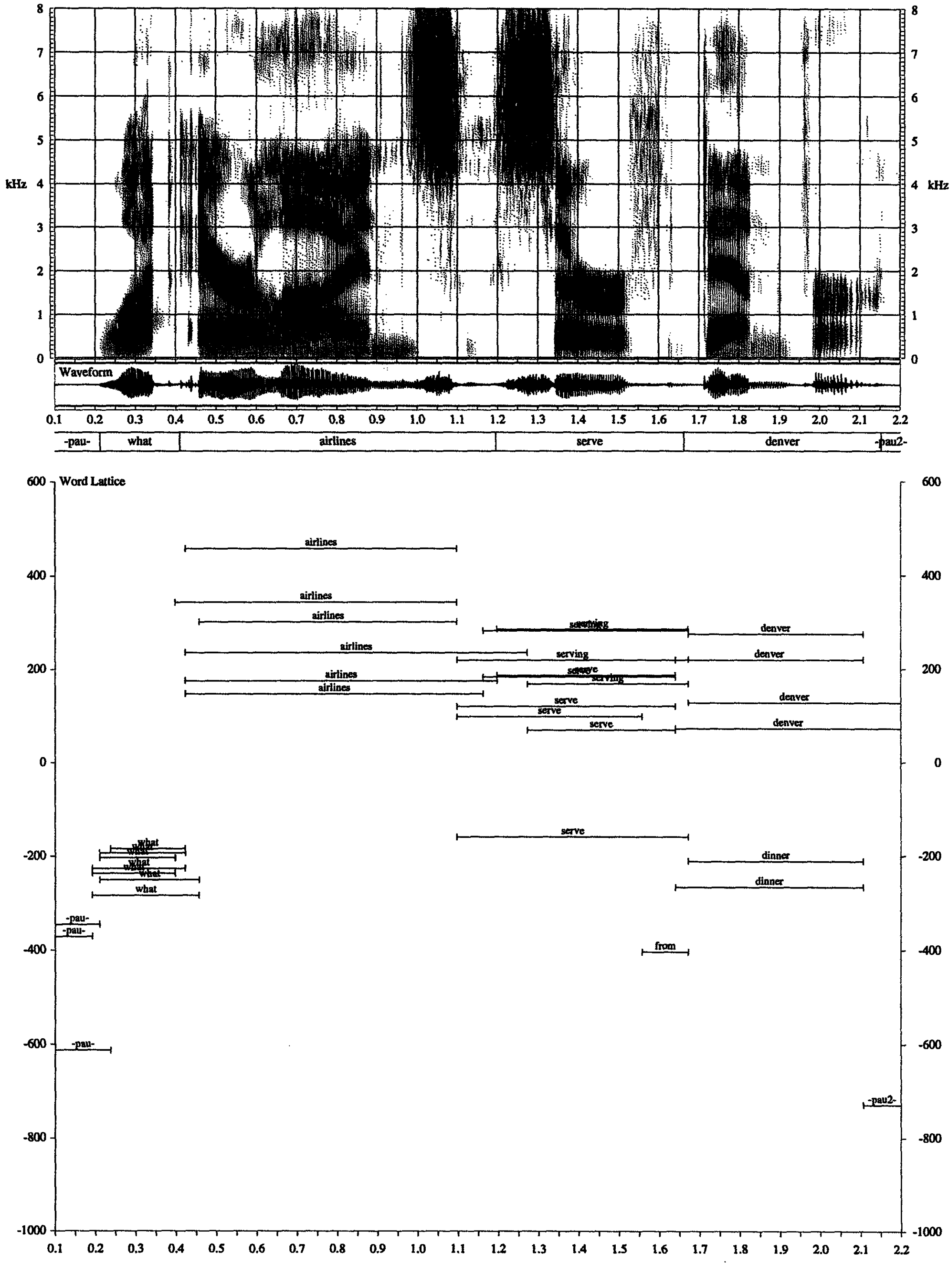


Figure 4-13: Word lattice. In the word lattice, the vertical axis represents acoustic/lexical score, and the horizontal axis represents time.

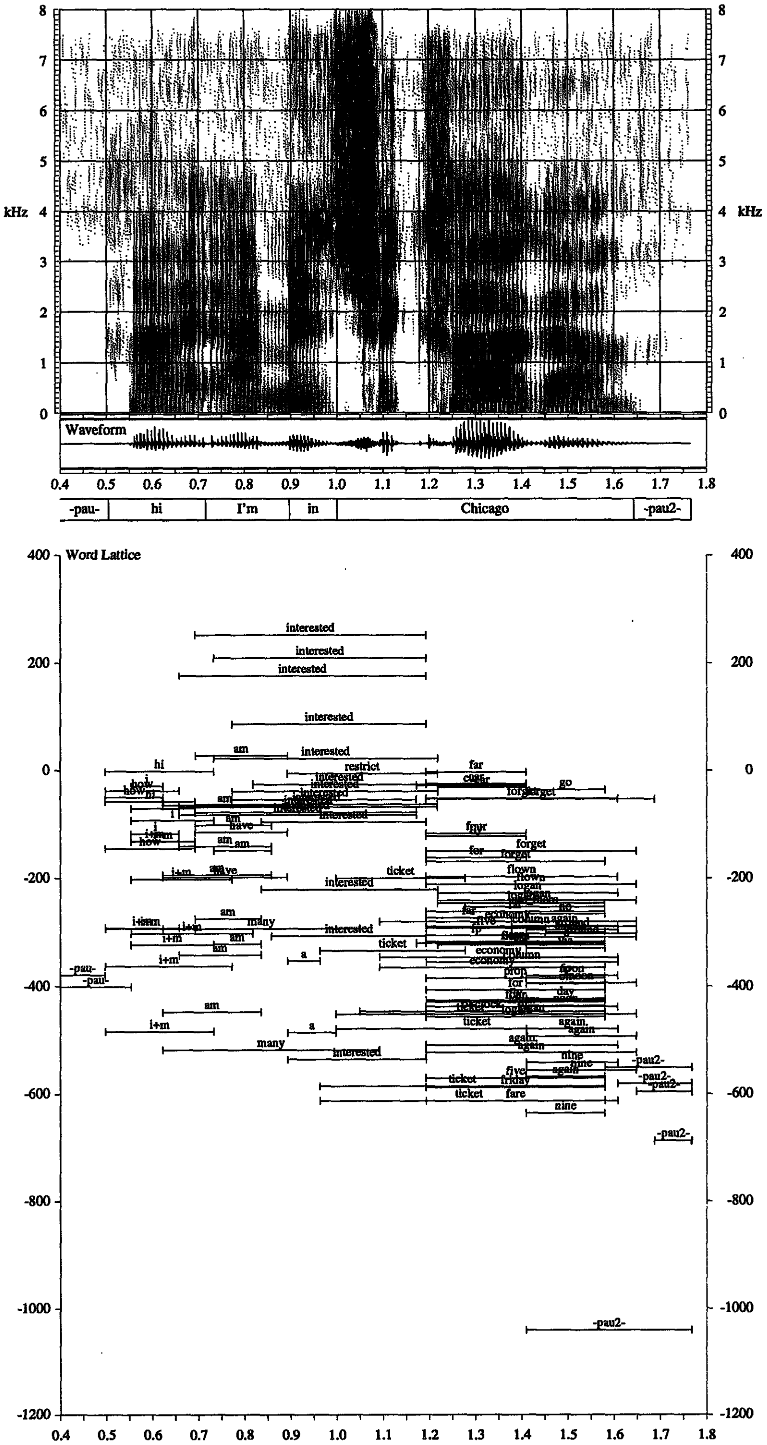


Figure 4-14: Word lattice for utterance containing an out-of-vocabulary word. In this example, the word "Chicago" is a new word and is responsible for a large number of word competitors. In the word lattice, the vertical axis represents the acoustic/lexical score, and the horizontal axis represents time.

Active-Word Counts

While word lattices can be helpful in displaying the *individual* competitor words and their scores, they can easily become too unwieldy to be useful, especially as the relative score threshold θ is increased. If we are more interested in the number of word competitors than we are in their identity, we can compute summary statistics. One type of statistic we have examined we call a time-slice statistic. The general idea behind a time-slice statistic is to count the number of word graph edges that cross every possible time slice.

We examined several different ways of counting edges and chose to count the number of distinct word labels that cross a particular time slice. If we only count the total number of edges across a given time slice, "jitter" in the endpoints of edges tends to increase the count dramatically. However, if we count the number of distinct words, edges that share the same word label are counted only once. We call the time-slice count of distinct words the *active-word count.* The active-word count has an intuitive interpretation: it is the number of words in the vocabulary "active" above the relative score threshold θ during recognition.

Figure 4-15 shows the active-word count for the same utterance as in the word lattice of Figure 4-13. Two active-word counts are plotted simultaneously for two different relative score thresholds θ (300 and 800) where the smaller threshold is the same one used in the word lattice. The vertical lines are the locations of the acoustic boundaries (see Section 3.2) and represent the maximum resolution along the time axis at which the counts can be made. In this example, in which there are no out-of-vocabulary words, we see relatively low counts, indicating that there are few distinct word competitors. Even at the much larger $\theta = 800$, the number of active words remains below 20.

In contrast, Figure 4-16 shows the active-word count for the same utterance as in the word lattice of Figure 4-14, which contains the out-of-vocabulary word "Chicago." In this example, we see a very large number of active words, nearly 320 or 25% of the vocabulary, in the region of the new word at the relative score threshold $\theta = 800$. Compared to Figure 4-15 the number of active words is significantly larger. Again, this difference is largely due to the presence of the new word "Chicago," where the recognizer

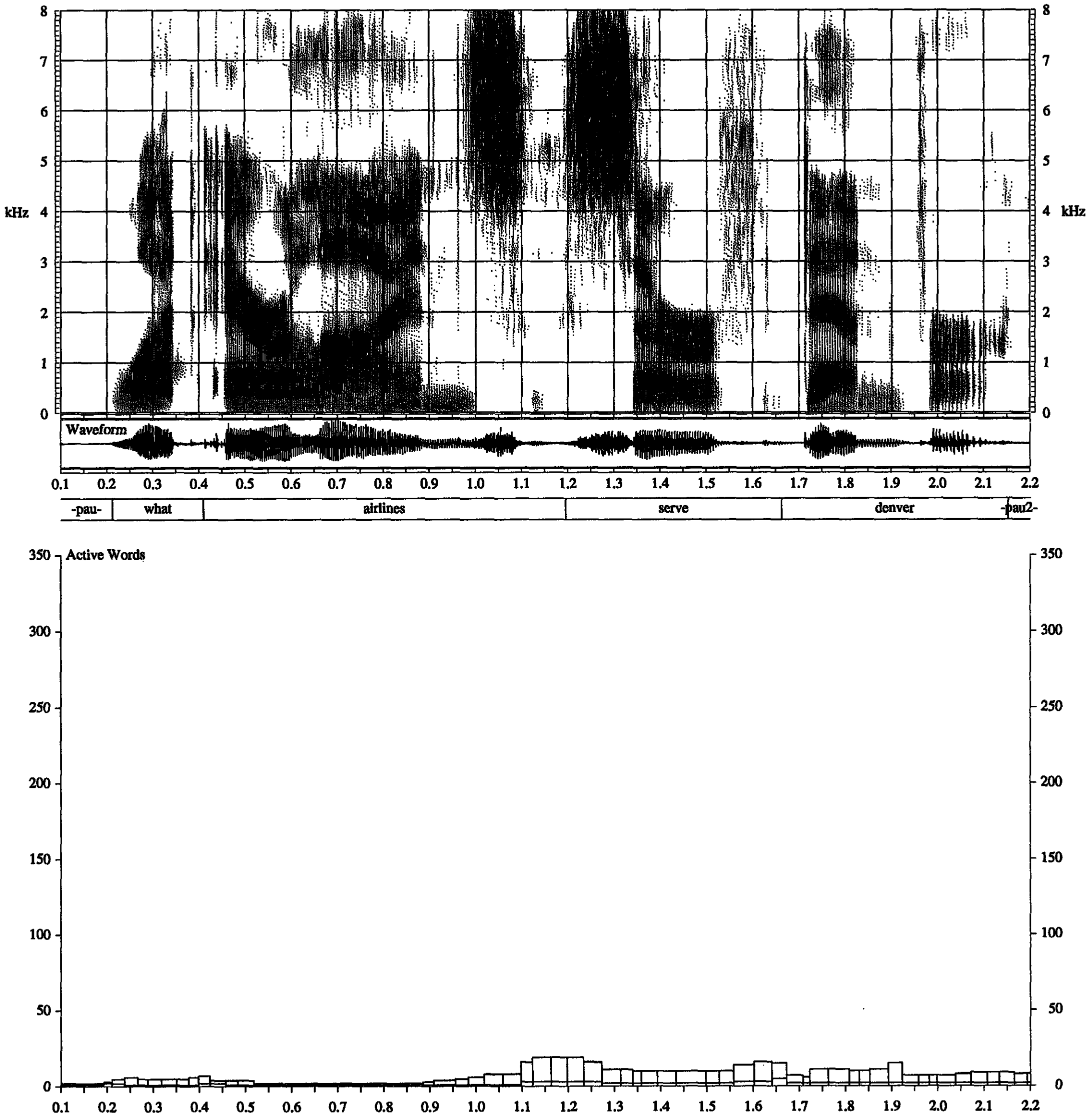


Figure 4-15: Active-word counts. In this example, all words are in-vocabulary. The active-word count is plotted for two different relative score thresholds θ (300 and 800).

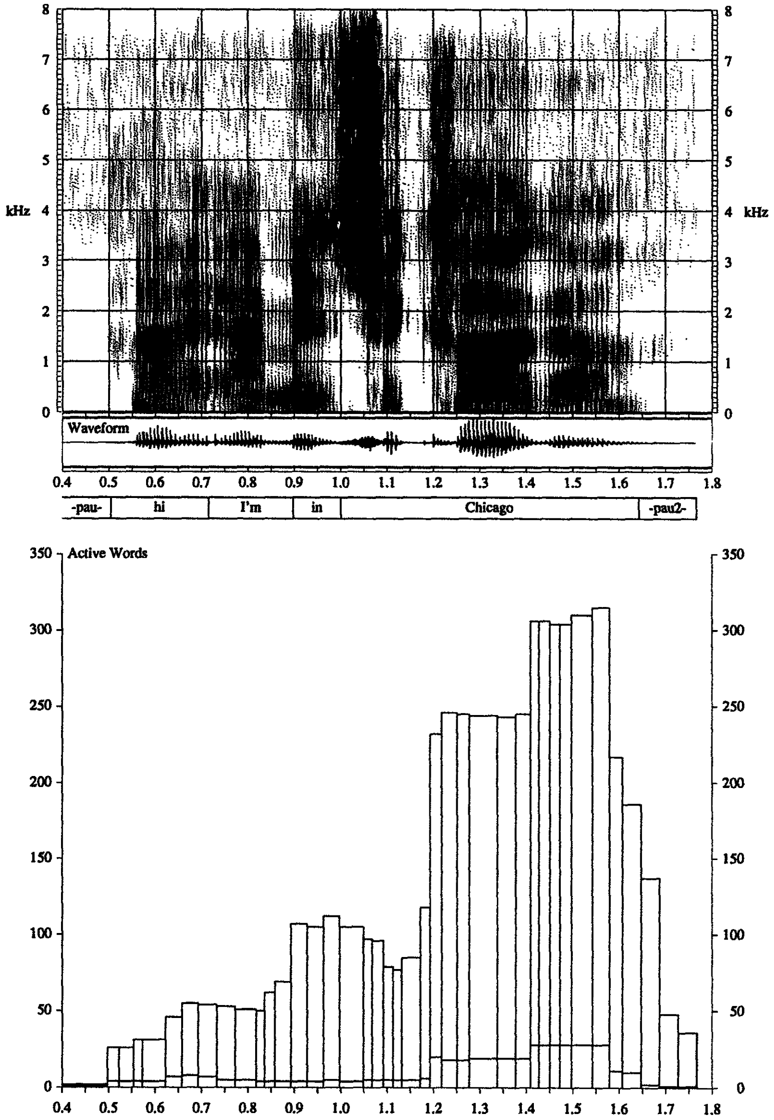


Figure 4-16: Active-word counts with an out-of-vocabulary word. In this example, the word "Chicago" is a new word and is responsible for a large number of word competitors. The active-word count is plotted for two different relative score thresholds θ (300 and 800). Note that the vertical scale is different in Figure 4-15.

has been forced to hypothesize a large number of words in an attempt to explain the acoustics of the new word. The active-word counts indicate recognizer "confusion" in the vicinity of the new word.

Later, in Section 5.4 we examine the contents of word graphs in order to study the effect that new words have on computation during the recognition search. We show empirically that new words significantly increase the number of edges contained in word graphs which implies that both the A* N-best and word graph searches explore a greater number of word hypotheses. We examine the impact that *position* of new words in different regions of an utterance (e.g., near the beginning or near the end) has on the computational demands of the search as measured by word graph complexity. Finally, we examine the active-word count in the vicinity of new words and show that it is indeed correlated with the location of new words, and hypothesize that it might be a useful measure during new-word detection. Not only do we post-process word graphs for all the recognition performance experiments of Chapter 5, we also used them to analyze computational demands due to new words.

4.5 Related Research

Ours is not the only system capable of generating word graphs or related representations. Researchers at Philips, SRI, and INRS-Télécommunications have all published algorithms for computing word graphs. Evidently, word graphs/networks/lattices are becoming an increasingly popular alternative to N-best lists. Judging by the dates of publication, all of this research, including our own, was performed in the same time period of 1992–1994. The development of our A* word graph algorithm, first reported in [27], was conducted independently of the other word graph research presented in this section.

Oerder and Ney [48] and Aubert et al. [7] at Philips present word graphs, similar to those we present in this chapter, which interface speech recognition and natural language components. Their algorithm also makes use of two passes in opposite directions. In the first pass forward in time, the "word hypotheses generator" produces word hypotheses that become edges in the graph. This pass is a minor extension of their normal Viterbi-style search where word hypotheses are incrementally added to a graph representation. In the second pass backward in time, the "word graph optimizer" prunes the graph created in the first pass by eliminating parts of the graph that contribute only to different alignments of the same word strings. This is accomplished by pruning word edges that do not belong to complete-utterance hypotheses that score above a specified threshold, and by merging subgraphs with identical labels and begin/end times.

The Philips word graphs appear to have fewer word edges as compared to our word graphs due to the pruning of different alignments. However, it is difficult to compare the two precisely since the Philips word graphs were analyzed on a different task. Furthermore, the score thresholds used in their system are undoubtedly different from ours, and they did not relate their word graph size to N-best list size. Overall, the word graph algorithm developed at Philips [7, 48] sounds promising. The fact that it can begin generating word edges in the first pass may mean that the first and second passes could run in parallel, whereas our second pass cannot begin until the first Viterbi pass is complete.

Murveit et al. at SRI [44] use a multi-stage search technique that produces word lattices as an intermediate representation. Their "forward-backward word-life" algorithm generates word lattices in forward and backward passes. The algorithm is similar to that used in the Philips system [7, 48] in that word edges are accumulated during the first Viterbi-style pass and then pruned in the second pass in the opposite direction. However, a critical difference between the SRI lattices and the word graphs of the Philips and SUMMIT system is that they do not store acoustic/lexical and language model scores in the lattice. Rather, they use their lattices as a word-transition "grammar" for subsequent search stages utilizing more detailed models. The word edges in the SRI word lattices contain only word labels and begin/end times.

Kenny et al. at INRS-Télécommunications [36] present a multi-pass approach to the speech recognition search problem that produces word graphs after three passes. In the first pass, backward in time, a phonetic graph is produced by using one- or two-phone look-ahead. In the second pass, forward in time, a word graph is produced by imposing lexical constraints and a coarse language model while searching through the phonetic graph. In the third pass, backward in time, the word graph of the second pass is pruned by performing an exhaustive traversal of the word graph and deleting edges whose complete-utterance score (i.e., f) is above a prescribed threshold. Finally, in an optional fourth pass, forward in time, detailed acoustic models and a fine language model are used to re-score hypotheses. The search in the last pass is exhaustive if the word graph is small enough. Otherwise, an A* search is used with an (inadmissible) heuristic stored in the graph.

Broadly speaking, Kenny et al.'s word graphs are very similar to ours. At the end of the second pass, their word graph contains all word edges whose A* score f^* is above a specified threshold. At the end of the third pass, their word graph has been pruned such that it contains all word edges whose score f (i.e., no longer a heuristic estimate f^*) is above the threshold. This is exactly the set of word edges our word graphs contain. Their fourth pass corresponds to the re-sorting the SUMMIT system performs when more detailed language and acoustic models are applied.

4.6 Summary

In this chapter we have presented a novel algorithm for computing word graphs using a two-stage search. This algorithm represents a minor change to the A* N-best algorithm used in the SUMMIT system. While we initially developed the word graph approach in order to study recognizer behavior (Section 4.4.2) we have found the approach to be useful in general. As a result, the word graph algorithm is now a part of the SUMMIT system and is used to interface the acoustic-lexical search component to a more detailed language modeling component. In this chapter we have demonstrated that the A* word graph search algorithm can be significantly more efficient, both in computation time and in representation size, than the more traditional A* N-best algorithm. Further, we have presented two display tools, based on word graphs, that can be used to examine recognizer behavior in the vicinity of new, out-of-vocabulary words.

Chapter 5

A Recognizer-based Study

In Chapter 2 we presented a recognition system–independent study of the new-word problem. While such an examination of the problem based on orthographic transcriptions reveals some important characteristics of new words, such as their lexical, syntactic, and phonological properties, it does not reveal how new words interact with a continuous-speech recognition system. In this chapter, we present a characterization of the new-word problem in the context of the SUMMIT speech recognition system. We examine two parts of the new-word problem:

1. the effect new words have on recognition performance and computation when there is no new-word detection capability, and
2. the relative importance of updating or retraining system components when adding new words to the system vocabulary.

The goal is to derive an empirical understanding of the problem of new words within a recognition system to complement the more abstract analyses of Chapter 2. We want to *quantify* the effects that new words have on recognition accuracy, search, as well as on acoustic, phonological, and language modeling. If we are to build useful spoken language systems, they must not only detect the presence of new words, but be able to incorporate them into a system vocabulary dynamically. In order to add new words to a system vocabulary, various recognition components must be updated. In our study, we will examine the relative importance of updating the acoustic models, lexical (pronunciation) models, and class n-gram language models. We will also discuss features of these components that enable them to be easily and effectively updated.

5.1 Methodology

First and foremost, we wanted our empirical recognizer-based study of the new word problem to be carefully controlled. We wanted to be able to separate the errors caused by the occurrences of new words from the errors that the recognition system would otherwise make. To be able make this distinction between new-word errors and system errors we needed to use a *baseline,* or control, system in our experiments. In order to control for system errors not related to new words, we needed evaluation utterances that were strictly in-vocabulary for the baseline system; the baseline system would not encounter new words. To evaluate the effect of new words, we needed a test system that did encounter new words. The way to accomplish this was to *simulate* new words for the test system by using a reduced vocabulary. The system performance comparisons were most controlled if the reduced vocabulary was a subset of the baseline vocabulary. Further, we tested both systems, the reduced-vocabulary system and the baseline system on the *same* utterances. The difference in performance of the two systems represented the effect of new words since system errors not due to new words were factored out by the baseline system.

We also wanted to evaluate the capabilities of our system to learn new words. The baseline system was helpful for this purpose too. We viewed the baseline system, which was fully trained on the set of simulated new words, as an upper bound on how well our system could "learn" new words. Thus, in our experiments to determine how well the reduced-vocabulary system's acoustic, lexical, and language models could incorporate new words (without training on them) we could use the baseline system as a yardstick. Thus, the shortfall between the performance of an updated version of the reduced-vocabulary system and the performance between the baseline system on the same utterances represented the difference between updating the various models without training and fully training those models on occurrences of the new words.

Given that we chose to simulate new words for our recognition study, we needed to choose a task and two vocabularies from which to build the baseline and reduced-vocabulary systems. The basic model we used in simulating new words was as follows:

1. there was an original system with limited vocabulary (the reduced-vocabulary system),
2. users generally stayed within the domain of the original system but did not know the exact limits of its vocabulary, and
3. an updated system with a larger vocabulary (baseline system) was built to handle the new words.

The ATIS domain is appropriate for this scenario. The original, reduced vocabulary could be represented by a vocabulary based on ATIS-2 utterances. When ATIS-3 was introduced, it represented an expansion of the task vocabulary, but the domain remained essentially unchanged. We treated the ATIS-3 utterances as input from users who did not know the exact limits (e.g., allowed cities, airports, and airlines) of the system. Thus, we could think of ATIS-3 utterances as being within the reduced-vocabulary system's domain but containing out-of-vocabulary words. Finally, we could build the baseline system with a larger vocabulary that covered the ATIS-3 vocabulary.

In examining the effects of new words on recognizer performance, we chose to use word-error rate as our performance measure [54]. To compute word-error rate, we aligned, word-for-word, recognizer output with reference orthographies. Once aligned, the number of word substitutions, deletions, and insertions can be measured. The total of these is the number of word errors; when converted to a fraction of total reference words, we arrive at the new-word rate. Figure 5-1 shows an example of string alignment with substitutions, deletions, and insertions indicated.

Figure 5-2 shows the three primary components of the SUMMIT system that we examined in our study of the issues related to learning new words: the acoustic models, the lexical (pronunciation) models, and the n-gram language models. For all of these models, we built small- and large-vocabulary versions by using different training sets with small and large vocabularies. To examine the relative importance of updating the

airline	with	code	f		f
airline		code	f	to	Denver
	d			*i*	*s*

Figure 5-1: Example string alignment. The reference (correct) string is at the top, and the recognizer's top hypothesis is at the bottom. Errors are marked with *s*, *d*, and *i* for substitution, deletion, and insertion, respectively. This example has a total of three errors out of five reference words, yielding a word-error rate of 3/5.

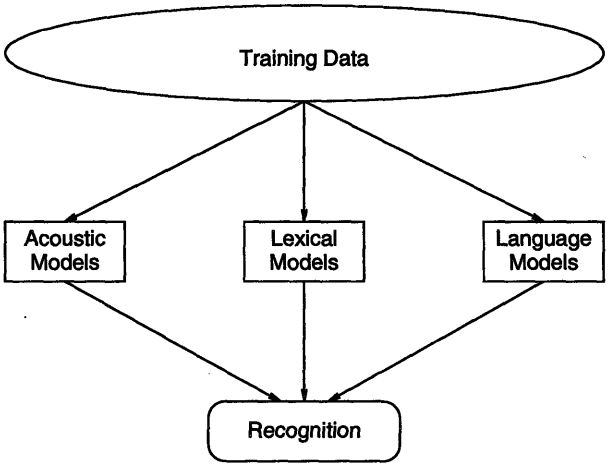


Figure 5-2: Primary recognizer components.

various components on new words, we systematically exchanged components trained on the small and large vocabularies. Since we could consider the baseline system as having fully "learned" the new words, it could serve as the control condition. By measuring the performance shortfall when different small-vocabulary components were installed we could deduce how important it was to train them when learning new words.

Overall, in the recognizer-based study of the new-word problem in this chapter we tried to conduct carefully controlled experiments in which the effects of new words were isolated from other system-dependent effects. We used two systems in our experiments, a baseline system for which all words were in-vocabulary and a reduced-vocabulary system that encountered new words. Further, experiments were conducted by comparing systems on the same set of utterances.

5.2 Recognizer Configuration and Training

In this section we describe how we determined the baseline and reduced vocabularies, how we prepared the training and testing data, how we trained the systems, and the performance of the baseline system.

5.2.1 Vocabulary Determination

To create the reduced-vocabulary system, we removed words from the larger vocabulary of the baseline system. In general, the large vocabulary was based on ATIS-3 utterances, and the small vocabulary was based on ATIS-2 utterances. This division is reasonable because ATIS-3 represented a natural extension of ATIS-2 to new cities, airports, and airlines. It allowed us to test a system with the smaller ATIS-2 vocabulary on ATIS-3 vocabulary and encounter new words due to the expansion of the task vocabulary. To a first-order approximation, these simulated new words model the new words that we might expect when users know about the ATIS domain, but not about the precise vocabulary (e.g., allowable set of cities, airports, and airlines). Therefore, when we tested the reduced-vocabulary system on a subset of the ATIS-3 data, it encountered a large number of simulated new words. (See Section 5.2.2 for a description of the test set.) Altogether, there were 18,191 ATIS-2 utterances and 8,392 ATIS-3 utterances available for our study.

The baseline system's large vocabulary was the vocabulary used by the SUMMIT system for the ARPA December 1993 ATIS-3 evaluation [77]. This vocabulary contained 2,461 words and was based on a vocabulary supplied by Carnegie Mellon University.

The reduced vocabulary was to be built from ATIS-2 utterances, and the words for it were extracted from the baseline ATIS-3 vocabulary using the following procedure. We counted the number of times each of the baseline-vocabulary words occurred in the ATIS-2 utterances. These word counts enabled us to determine which words were justifiable based on ATIS-2 utterances. Words that did not occur in ATIS-2 were removed unless they could be easily derived from a word that did occur (e.g., the word "faster" would have been retained if "fast" occurred). Additionally, cities, airports, and airlines

Atlanta	Dallas	Fort Worth	Pittsburgh
Baltimore	Denver	Oakland	San Francisco
Boston	Detroit	Philadelphia	Washington

(a) Cities in reduced vocabulary (ATIS-2)

Burbank	Las Vegas	New York	St. Louis
Charlotte	Long Beach	Newark	St. Paul
Chicago	Los Angeles	Ontario	St. Petersburg
Cincinnati	Memphis	Orlando	Tacoma
Cleveland	Miami	Phoenix	Tampa
Columbus	Milwaukee	Salt Lake City	Toronto
Houston	Minneapolis	San Diego	Westchester County
Indianapolis	Montreal	San Jose	
Kansas City	Nashville	Seattle	

(b) Additional cities in baseline vocabulary (ATIS-3)

Table 5-1: Cities in ATIS-2 and ATIS-3.

that were explicitly added during the ATIS-2 to ATIS-3 expansion were removed. Altogether, 1,135 words were removed from the baseline vocabulary to yield the reduced vocabulary of 1,326 words.

The fundamental difference between the reduced and baseline vocabularies lies in the number of cities represented. Table 5-1 lists the cities contained in the reduced vocabulary and the additional cities contained in the larger baseline vocabulary. The reduced vocabulary contains 11 cities and 9 airports, versus the baseline vocabulary's 46 cities and 52 airports.

Because we modeled the simulated new words on the difference between the underlying ATIS-3 and ATIS-2 vocabularies, most of the simulated new words were the names of cities, airports, states, and airlines associated with the explicit task expansion. Table 5-2 lists a random sampling of the other new words that were not associated with the expansion of ATIS but were still excluded from the reduced vocabulary because they did not occur in ATIS-2.

adjectives	acceptable, alternative, comfortable, red-eye, surface, worse
nouns	bagel, fact, friend, home
verbs	assuming, clarify, fitting, suggest

Table 5-2: Sample new words not associated with explicit ATIS expansion.

5.2.2 Testing Sets

In order to have an adequate number of new-word occurrences for our studies, our test utterances came from ATIS-3. Specifically, we combined the ATIS-3 development and test sets, 1,737 utterances total, to form our testing sets. Of these utterances, 261 contained spontaneous speech events other than pauses. We discarded these utterances because the problem of partial words in spontaneous speech was beyond the scope of this thesis. Of the remaining 1,476 utterances, 27 contained out-of-vocabulary words for the baseline system. We discarded these 27 utterances because we wanted the entire test set to be in-vocabulary for the baseline system so that it could be used as a control in our experiments. Altogether, this left 1,449 utterances (12,707 words) in the testing set $\mathcal{S}$ that were strictly in-vocabulary for the baseline system.

The new-word rate measured over $\mathcal{S}$ was 12.7%. According to the results of Chapter 2, we would expect a new-word rate of about 1% for a 1,300-word vocabulary in the ATIS domain. The reason our new-word rate was so much higher was due the *artificial* expansion of the ATIS task between ATIS-2 and ATIS-3. The high new-word rate is advantageous for our purposes because we need a large number of new words to study the effects of new words.

In order to distinguish the effects of zero, one, and multiple new words per utterance, we further subdivided the test set $\mathcal{S}$. Table 5-3 summarizes the sizes and number of new words in the subsets $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_{2+}$. $\mathcal{S}_0$ contains zero new words, $\mathcal{S}_1$ contains exactly one new word per utterance, and $\mathcal{S}_{2+}$ contains two or more new words per utterance, with $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_{2+}$. As the number of new words per utterance increases, so does the average number of words per utterance.

All test sets were independent of the training sets. Further, all test-set speakers were different from the train-set speakers to ensure speaker-independent recognition results.

	utterances	words	words/ utterance	new words	new words/ utterance
$\mathcal{S}$	1,449	12,707	8.8	1,618	1.1
$\mathcal{S}_0$	684	4,819	7.1	0	0.0
$\mathcal{S}_1$	198	1,656	8.4	198	1.0
$\mathcal{S}_{2+}$	567	6,232	11.0	1,420	2.5

Table 5-3: Testing sets.

	utterances	words
baseline (full)	20,397	172,555
reduced-vocabulary	16,953	139,513
baseline (control)	16,891	143,068

Table 5-4: Training sets.

5.2.3 Training Sets

We made use of three overlapping training sets: the largest training set for the baseline system, a smaller training set for the reduced-vocabulary system, and another small training set for a baseline-vocabulary system to be used as a control for training set size. All training sets were based on the combination of all (train, development, and test) ATIS-2 utterances plus the ATIS-3 train set. Altogether, this yielded 22,427 utterances containing 218,615 words. However, a number of these utterances contained words not in the baseline vocabulary, many of them due to partial words from spontaneous speech. These utterances which contained new words for the baseline system were discarded.

SUMMIT's training process discards all utterances containing out-of-vocabulary words. Therefore, the total number of utterances used to train the baseline system was reduced from 22,427 utterances to 20,397 utterances. When training the reduced-vocabulary system, additional utterances were discarded due to the increased number of out-of-vocabulary words. Finally, as a control to ensure that performance differences were not due to amount of training data used, we created a baseline-vocabulary training set roughly the same size as the reduced-vocabulary training set by randomly sampling the utterances in the baseline training set. The sizes of the three training sets are summarized in Table 5-4.

5.2.4 Training Acoustic and Lexical Models

The SUMMIT system uses an iterative process to train the acoustic-phonetic and lexical (pronunciation) models. The reason for the iteration is twofold:

1. the training utterances were not phonetically transcribed, so the system iteratively improves its own transcription (i.e., forced alignment) of the utterances; and

2. the training of the weights in the pronunciation graphs, the lexical weights, requires a number of iterations in order to converge.

Training the acoustic models requires forced alignment[1] paths that essentially contain time-aligned phonetic transcriptions produced by the system itself. The acoustic features for each of the training tokens were collected and input into the mixture-model training algorithm; a maximum of 64 mixtures per phonetic unit were computed. Training the lexical weights requires both a forced and a best alignment[2] of each training utterance, which are compared, and lexical arcs that contribute to recognition errors are penalized. Penalizing lexical arcs in SUMMIT is necessary because application of the phonological rules can over-generate arcs, contributing to recognition errors.

The baseline system was the context-*independent* part of the SUMMIT system used in the ARPA December 1993 evaluation. This system underwent many training iterations. The iterative training of the lexical weights in the pronunciation models is the most computationally expensive part of training SUMMIT due to the computation of all the forced and best recognition paths. The baseline system was trained on the full training set of 20,397 utterances.

Some of our experiments required acoustic models that were not trained on utterances containing any of our simulated new words. We accomplished this by computing forced-alignment paths using the baseline system for all 16,953 of the utterances in the reduced-vocabulary training set. New acoustic models, containing no statistics from the the baseline models, were trained using these forced forced alignments. Therefore, the

[1] Forced alignment involves finding the optimal alignment of a reference string versus the waveform.

[2] The best alignment involves finding the alignment of the best-scoring hypothesis. This hypothesis is not necessarily the same as the reference string due to recognition errors.

reduced-vocabulary acoustic models were trained in the absence of the simulated new words.

Because we felt that the reduction of the number of training utterances could affect the results of our experiments using the reduced-vocabulary acoustic models, we trained another set of acoustic models using a comparably sized subset of the large-vocabulary training data. These models were similarly trained using a single iteration on the smaller baseline-vocabulary training set of 16,891 utterances. Thus, these models could be used as a control for training-set size.

5.2.5 Training n-Gram Language Models

The first step in training the class n-gram language models was to determine the word classes. Because our baseline system was the SUMMIT system used in the ARPA December 1993 evaluation, we used the classes developed for that version of the system. Altogether there were 53 classes, including day names, month names, state names, city names (including airport names), airline names, airline codes, and numbers. Not all words in the vocabulary belonged to classes. In the baseline vocabulary, only 781 words out of 2,461 belonged to one of the pre-determined classes. The remaining 1,680 words, in effect, belonged to their own class for the purposes of language modeling.

When adapting the SUMMIT system from the ATIS-2 vocabulary to the ATIS-3 vocabulary for the ARPA evaluation, SUMMIT developers determined empirically that optimal performance on ATIS-3 test material was achieved when ATIS-2 and ATIS-3 training texts were combined in a 1:3 ratio. For our experiments, we used this same ratio by duplicating ATIS-3 material three times.

Training of the n-gram language models entailed accumulating counts for all word sequences up to length n. These counts were turned into class-conditional probabilities with various types of smoothing as described in Section 3.6 on page 66.

5.2.6 Word Graphs

We utilized word graphs in our recognition experiments to interface the lexical access search (using bigram constraints) to the class n-gram re-sorting component. The word graphs were computed using the A* word graph search algorithm developed for this thesis and described in Chapter 4. The word graphs proved to be an effective tool for interfacing the higher-order class n-grams. These word graphs allowed us to quickly rerun recognition experiments with different n-gram language models without having to redo the acoustic and lexical modeling.

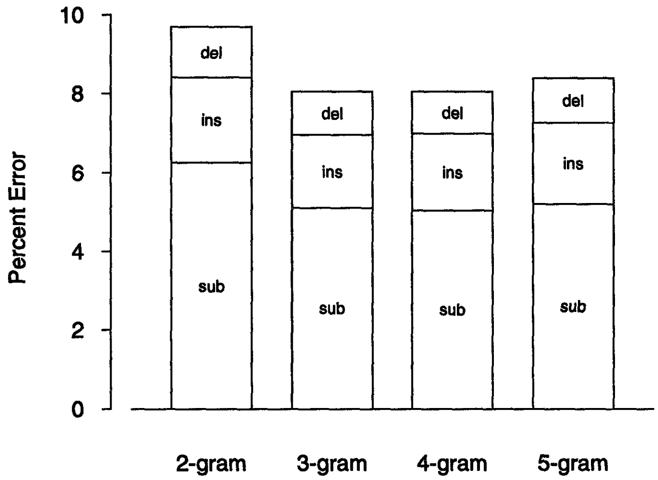
We chose to use a relative score threshold $\theta = 800$ for our experiments, meaning that our word graphs contained all N-best hypotheses, and all their alignments, that scored within 800 of the best-scoring hypothesis. By itself, $\theta = 800$ is not very meaningful. What is important is the fraction of correct hypotheses contained in the word graphs. Correct hypotheses that were not included in the word graphs were errors that could not be corrected with improved language modeling during re-sorting.

We computed maximum achievable sentence and word accuracies by searching through the word graphs using the correct reference word strings as constraints (i.e., a kind of forced alignment). We found that for $\theta = 800$, fully 97.2% of the test-set utterances contained the correct word string for the full utterance in the word graph. Further, we found that the minimum achievable word-error rate was only 0.5%. We deemed this error rate to be acceptable. It could not be significantly reduced for $\theta > 800$.

5.2.7 Baseline Performance

Figure 5-5 shows the performance of the baseline system on the full test set $\mathcal{S}$ for various class n-gram language models in the range $2 \leq n \leq 5$. The most notable increase in performance resulted from using a class 3-gram (trigram) instead of a class 2-gram (bigram) language model. The performance increase was due to the better constraining power of the 3-gram as evidenced by the decrease in perplexity L. The performance and perplexity differences between the 3-gram and 4-gram models were negligible, and for the 5-gram both were slightly worse,[3] presumably because of sparse data problems. Particular word/class sequences of length 5 were too sparse in the training text to yield reliable probability estimates. Even though the 3-gram and 4-gram performed almost

[3] The difference between the total number of word errors for the 4-gram and 5-gram models is not significant at the 0.05 level.



(c)

language model	*L*	*sub* *n*	*sub* *%*	*ins* *n*	*ins* *%*	*del* *n*	*del* *%*	*total* *n*	*total* *%*
2-gram	28.64	795	6.3	274	2.2	161	1.3	1,230	9.7
3-gram	18.88	648	5.1	234	1.8	142	1.1	1,024	8.1
4-gram	18.55	640	5.0	247	1.9	136	1.1	1,023	8.1
5-gram	18.60	661	5.2	262	2.1	144	1.1	1,067	8.4

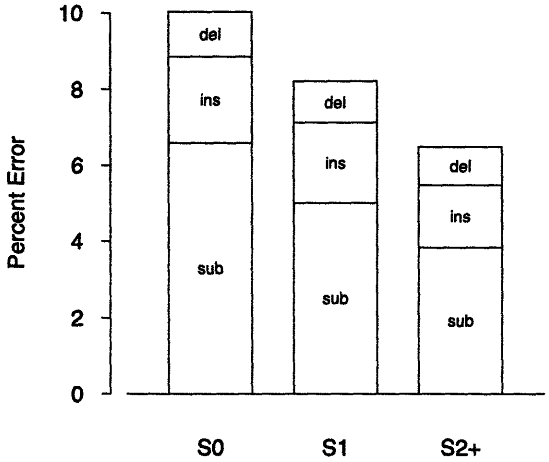
(d)

Figure 5-5: Baseline performance with different language models. Perplexity L and performance measured over entire test set $\mathcal{S}$ of 12,707 words (1,449 utterances). For each class of error (i.e., substitution, deletion, insertion, and total) the number and percentage of errors is given. Total represents the total word-error rate.

identically, we chose to use the class 4-gram for our experiments because the SUMMIT system typically uses the class 4-gram for N-best re-sorting.

Figure 5-6 shows the performance of the baseline system (with class 4-gram language model) on the testing subsets $\mathcal{S}$, $\mathcal{S}_1$, and $\mathcal{S}_{2+}$.[4] It is interesting to note that accuracy on $\mathcal{S}_{2+}$ was better than on $\mathcal{S}_1$, which was better than on $\mathcal{S}_0$. We divided the string-aligned system output into two sets: those words aligned with reference words that are simulated new words (that are in-vocabulary for the baseline system) and those that

[4] Recall that the subsets were defined for new words not in the *reduced* vocabulary. The baseline system faced no new words in any of the testing subsets.



(a)

		sub		*ins*		*del*		*total*	
set	n_{total}	*n*	%	*n*	%	*n*	%	*n*	%
$\mathcal{S}$	12,707	640	5.0	247	1.9	136	1.1	1,023	8.1
$\mathcal{S}_0$	4,819	318	6.6	109	2.3	56	1.2	483	10.0
$\mathcal{S}_1$	1,656	83	5.0	35	2.1	18	1.1	136	8.2
$\mathcal{S}_{2+}$	6,232	239	3.8	103	1.7	62	1.0	404	6.5

(b)

Figure 5-6: Baseline performance on testing subsets. These baseline results were obtained with the class 4-gram language model.

are non-new words. We found that the overall non-new-word error rate was 8.5% versus 5.3% for the set of simulated new words. One possible explanation for the reason that baseline performance on $\mathcal{S}_1$ and $\mathcal{S}_{2+}$ is better than on $\mathcal{S}_0$ is that those sets contain a greater number of simulated new words, which the baseline system is better able to recognize (compared to non–simulated new words).

5.3 Recognizer Performance in the Face of New Words

In the study of new words presented in Chapter 2 we demonstrated that new words occur in a wide-variety of tasks no matter how large a system vocabulary is used. However, we did not relate the occurrence of new words to recognition errors within a continuous-speech recognition system. In this section we examine what happens to recognition performance when a system encounters new words. Certainly, a typical system is guaranteed to misrecognize a new word since it has no way of modeling it or knowing its orthography. However, it is likely that the occurrence of a new word could have a ripple effect and cause other words in an utterance, that are in-vocabulary, to be misrecognized due to contextual effects.

One thing we examined was what a recognizer proposes in the place of a new word (i.e., what it substitutes from its vocabulary). We were curious to see if the substitutions were acoustically and/or linguistically plausible. We called recognition errors due to substitutions for new words *direct new-word errors.* Perhaps of more importance, we examined the errors caused by new words on other, in-vocabulary parts of utterances. We called these errors *indirect new-word errors.* If recognition errors are limited to the new words themselves, then most of an utterance containing a new word will be correctly recognized. In examining indirect new-word errors, we divided them up into errors before new words, after new words, and between new words to see if relative position of a new word affected the rate of indirect new-words. In terms of both direct and indirect new-word errors, we examined performance on utterances that contained two or more new words to see if multiple new words present special problems.

In our performance analyses, we always compared the output of the reduced-vocabulary system with that of the baseline system in order to factor out recognition errors the system would otherwise make. Further, we always ran the reduced-vocabulary and baseline systems on the same set of utterances.

5.3.1 Error Analysis Methodology

As previously mentioned, we evaluated recognizer performance using word-error rates (by measuring the number of substitutions, deletions, and insertions). To measure these errors, recognizer outputs were aligned word-for-word with correct reference strings.

The general method of aligning hypothesized and reference word strings involves finding the alignment with the lowest cost. Typically, substitutions, deletions, and insertions are all of equal cost (e.g., one per error). There may be more than one alignment between two strings that have the same total cost, and the alignment process arbitrarily chooses one. The net result is an alignment with minimal *total* cost (i.e., sum of substitutions, deletions, and insertions).

However, in our analysis of the effects new words have on recognizer performance we were interested not only in total errors, but in errors aligned directly with new words and errors within in-vocabulary regions of utterances. Thus, we were concerned not only with alignments of minimal total cost but also with alignments that spread the cost appropriately between new words and non-new words. In other words, we were also concerned with the *quality* of alignment and not just total alignment score.

To improve alignment quality when new words were present, we modified the alignment costs. For our special new-word alignment, we set the cost of substitutions, deletions, and insertions to zero when the reference word was a new word and set the costs to one otherwise. The effect of this change was to align as many of the recognition errors as possible with new words, thus maximizing the alignment score over the in-vocabulary regions. Since a recognizer can propose a number of in-vocabulary words in the place of a new word, we would like those recognition errors to be associated with the new word and not with the rest of the utterance. Our special provisions for alignment near new words tended to associate insertions adjacent to new words with the new words.

Figure 5-7 shows an example of our special treatment of new words during alignment. In Figure 5-7(a), the alignment is the normal alignment. In Figure 5-7(b), the alignment is the special alignment for new words. In the latter case, all errors in the example are aligned with the new word "Chicago." This special alignment is appropriate in that the hypothesized words "do you have no" were substituted for "Chicago" and not for words earlier in the utterance. The special alignment algorithm only affects the *allocation* of errors between new and non-new words. We found that it does make a significant difference, especially when several words are substituted for a single new word. In such cases, the normal alignment procedure allocates several insertions to the non-new words as evident in Figure 5-7(a), when they should be allocated to the new word. Intuitively, the right thing to do is to maximize non-new-word performance by allocating as many errors as possible to the new words (i.e., minimizing the number of

I	want	to	arrive	in				*Chicago*	around	seven	p	m
I	want	to	arrive	in	do	you	have	no	around	seven	p	m
					i	*i*	*i*	*s*				

(a) Normal alignment

I	want	to	arrive	in	*Chicago*				around	seven	p	m
I	want	to	arrive	in	do	you	have	no	around	seven	p	m
					s	*i*	*i*	*i*				

(b) Special new-word alignment

Figure 5-7: Example of special new-word string alignment. The word *Chicago* is a new word. The vertical lines delimit the errors that are associated with the new word. In (a), there three errors associated with in-vocabulary (other) words, and one error associated with out-of-vocabulary (new) words. In (b), there are no errors associated with other words, and four errors associated with new words. We consider the alignment in (b) to be more accurate in terms of its allocation of errors.

indirect new-word errors at the expense of direct new-word errors). In the recognition performance experiments in this chapter, we only used the special alignment when the recognizer actually faced new words. Specifically, we never used the special alignment with the baseline system since it never encountered out-of-vocabulary words.

In our performance analysis, we divided utterances into intervals based on the position of new words. Table 5-5 defines the utterance divisions we used. For example, in the utterance of Figure 5-7(b), errors aligned with the words "I want to arrive in" would have been classified as *non-new* and *before*, errors aligned with the word "Chicago" were classified as *new*, and errors aligned with the words "around seven p m" would have been classified as *non-new* and *after*. Additionally, in an utterance with multiple new words, there could be words classified as *non-new* and *between*. (In this context, "new word" means a word not in the *reduced* vocabulary.) Therefore, we could compute the performance on the set of "new words" for the large-vocabulary baseline system even though all words were in vocabulary. The baseline performance over the set of the small system's new words could thus be used as a control.

overall	all words
new	all out-of-vocabulary words
non-new	all in-vocabulary words
before	in-vocabulary words occurring before new words
between	in-vocabulary words occurring between new words
after	in-vocabulary words occurring after new words
adjacent	in-vocabulary words immediately adjacent to new words

Table 5-5: Regions of utterances for error analysis. In general, we performed recognition on entire utterances and divided the utterances after they were aligned.

5.3.2 Direct New-Word Errors

We first examined errors that were directly associated, or aligned, with new words. The goal was to characterize what the recognizer hypothesized in place of new words. It is important to understand the nature of the substitutions for new words because they could have a significant effect on *understanding.* In general, a recognizer will hypothesize a sequence of in-vocabulary words in the place of a new word. If the sequence of substituted words makes no sense syntactically and/or semantically, then the natural language component may have trouble interpreting the utterance hypothesis. If, on the other hand, the recognizer simply hypothesizes another word from the same word class as the new word, the natural language component may be able to interpret the recognizer's hypothesis, but will interpret it incorrectly. The latter case may cause more system and user confusion because neither may be immediately aware of the new word and the substitution made for it.

To examine the errors directly attributable to new words, we compared the output of the baseline system to a reduced-vocabulary system on $\mathcal{S}_1$, using a class 4-gram language model. We chose $\mathcal{S}_1$ because it contained only a single new word per utterance, thus simplifying our analysis. Further, to make as controlled a comparison as possible, the *only* difference between the baseline and reduced-vocabulary system was the vocabulary (i.e., the set of words that could be hypothesized). Both systems shared the same acoustic, lexical, and language models. When aligning recognizer hypotheses to reference strings, we used the normal alignment process for the baseline system and the special new-word alignment process described in Section 5.3.1 for the reduced-vocabulary system.

		sub		*ins*		*del*		*total*	
	n_{total}	n	%	n	%	n	%	n	%
baseline	198	16	8.1	2	1.0	0	0.0	18	9.1
reduced		182	91.9	14	7.1	64	32.3	260	131.3
Δ		+166	+83.9	+12	+6.1	+64	+32.3	+242	+122.2

Table 5-6: Direct new-word errors. The errors are measured for words aligned directly with the new words in $\mathcal{S}_1$, the set of utterances containing exactly one word each. Special new-word alignment was used for the reduced-vocabulary system's output. The total number of reference words over which word errors were measured is indicated by n_{total}. On average, there were 1.22 (242/198) direct errors per new word.

Table 5-6 shows the errors the baseline system made on the new words in $\mathcal{S}_1$ and the increase in errors made by the reduced-vocabulary system. Not surprisingly, most of the errors were word substitutions in which the system was forced to substitute an in-vocabulary word for a new word. Next in frequency were insertions, which resulted when the system substituted *multiple* words for a particular new word. Relatively few new words were deleted.

Table 5-7 shows a random sample of the reduced-vocabulary system's hypotheses for a few of the new words. Because the recognizer attempted to satisfy both acoustic and language model constraints, the recognition errors sometimes appeared to be acoustically reasonable and and other times reasonable from the point of view of the class n-gram language model, but rarely both. In many cases a word from the same (semantic) word class as the new word was substituted. In our experiments, fully 45% of new words experienced such within-class substitutions. These cases would be more difficult to detect because the natural language component would not notice an error and thus would interpret the incorrect utterance hypothesis. In the cases where the substituted words do not make sense syntactically and/or semantically the natural language component would likely have difficulty interpreting the recognizer's hypothesis.

The bottom line is that a recognizer is forced to propose in-vocabulary words in place of new words if it has no capability for detecting new words. The errors directly aligned with new words, direct errors as we call them, often include substitutions of multiple in-vocabulary words per new word. In some cases, the acoustics appear to dominate the selection of substitutes, while in others the language model appears to dominate, sometimes substituting a word in the same (semantic) class as the new word.

new word	→	*substitution*
Indianapolis	→	give me the list
Miami	→	Y N mean
Charlotte	→	show it
Seattle	→	several
Alaska	→	ask
Houston	→	interested
Cincinnati	→	the night
Burbank	→	very
Cleveland	→	Oakland
Kansas	→	Dallas
Alaska	→	US Air
Milwaukee	→	Oakland
O'Hare	→	Denver
Charlotte	→	Atlanta

Table 5-7: Sample hypotheses for new words. This table shows a randomly selected set of new-word substitutions made by the reduced-vocabulary system, with the new word at the left and the hypothesized word(s) at the right. We divided the hypotheses roughly into two categories: those that appear to be acoustically related and those that appear to be semantically related.

Overall, in testing set $\mathcal{S}_1$, which contains utterances with exactly one new word each, the reduced-vocabulary recognizer suffered 1.22 word errors per new word.

5.3.3 Indirect New-Word Errors

Of course, new words cause not only the errors associated with their own misrecognition, but also errors to in-vocabulary words in an utterance due to contextual effects. These indirect new-word errors, as we call them, are an important part of the new-word problem.

There are three primary reasons why in-vocabulary words can be misrecognized in the presence of new words:

1. lack of obvious word boundaries in continuous speech;

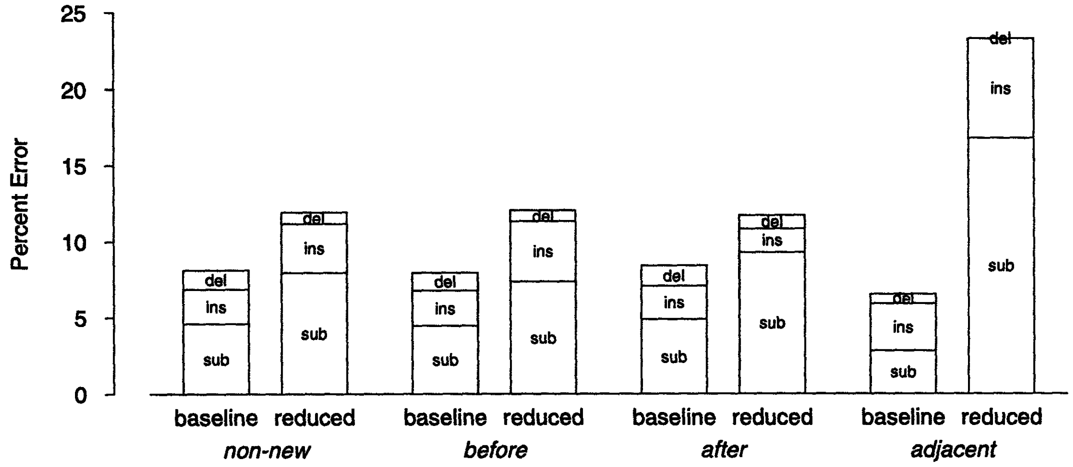
2. acoustic context due to co-articulation; and

3. language context due to language model constraints.

In the first case, the best way the recognizer might be able to explain the acoustics of a new word may be with *part* of an in-vocabulary word. If the recognizer hypothesizes a word longer than the new word, the recognition of neighboring words could be affected. In the second case, the boundaries of the new word may be respected by the recognizer, but phonemes near the word boundaries could affect context-dependent acoustic modeling. However, in our experiments with the SUMMIT system, we used context-*independent* acoustic models, and thus would not expect to see this effect. Finally, in the third case, words that the recognizer proposes in place of the new word can affect the language model scores for other words in the utterance. In the case of n-gram language models, we would expect this effect to be limited to words within a distance of $n-1$ words. However, because a recognition error could cause an adjacent in-vocabulary word to be misrecognized, words further than $n-1$ could be affected in a ripple effect due to the context dependence of language model scores. In the case of more sophisticated language models capable of enforcing long-distance constraints, words far from new words could be affected. Thus, due to contextual effects, the new word itself may not be the only word misrecognized in an utterance.

We analyzed the effect of new words on in-vocabulary regions of utterances, again by comparing the performances of a baseline system and a reduced-vocabulary system on test set $\mathcal{S}_1$. In fact, this analysis used the same utterances and aligned hypotheses as the direct new-word error analysis of the previous section. The only difference was in the regions of the utterances we examined. We divided the indirect new-word errors into different categories: all non-new words (*non-new*), in-vocabulary words preceding new words (*before*), in-vocabulary words following new words (*after*), and in-vocabulary words immediately before and after new words (*adjacent*). As in the previous section, recognition was performed using the class 4-gram language model.

Figure 5-8 summarizes the indirect new-word errors we obtained over different parts of $\mathcal{S}_1$ utterances.Over all non-new words, the word-error rate increased by 3.8% (a factor of 1.5). The increase in word-error rate due to indirect errors was comparable for the words before and after new words. The most notable feature of Figure 5-8 is that the set of words most affected by indirect new-word errors was the *adjacent* set, in which the word-error rate increased by 16.8% (a factor of 2.6). The vast majority of the errors



(a)

	n_{total}	*sub*		*ins*		*del*		*total*	
		n	%	n	%	n	%	n	%
non-new									
baseline	1,458	67	4.6	33	2.3	18	1.2	118	8.1
reduced		116	8.0	47	3.2	11	0.8	174	11.9
Δ		+49	+3.4	+14	+1.0	−7	−0.5	+56	+3.8
before									
baseline	1,006	45	4.5	23	2.3	12	1.2	80	8.0
reduced		74	7.4	40	4.0	7	0.7	121	12.0
Δ		+29	+2.9	+17	+1.7	+5	+0.5	+41	+4.1
after									
baseline	452	22	4.9	10	2.2	6	1.3	38	8.4
reduced		42	9.3	7	1.6	4	0.9	53	11.7
Δ		+20	+4.4	−3	−0.7	−2	−0.4	+15	+3.3
adjacent									
baseline	322	9	2.8	10	3.1	2	0.6	21	6.5
reduced		54	16.8	21	6.5	0	0.0	75	23.3
Δ		+45	+14.0	+11	+3.4	−2	−0.6	+54	*+16.8*

(b)

Figure 5-8: Indirect new-word errors. *Non-new* is measured over all in-vocabulary words in S_1, *before* over all words before new words, *after* over all words after new words, and *adjacent* over all words immediately adjacent to new words. By far, the adjacent words experience the most significant performance degradation. The total number of reference words over which word errors were measured is indicated by n_{total}. Altogether there were 198 new words in S_1, yielding an average of 0.28 (56/198) indirect errors per new word.

due to adjacent new words were substitutions.[5]

Both acoustic and language model contextual effects are responsible for the indirect new-word errors we observed. Acoustic contextual effects are generally fairly local (e.g., within a couple of phonemes). In contrast, language model constraints extend for several words. Therefore, we were curious to see if the length of language model constraints had a significant effect on the number of indirect errors due to new words. We ran the same experiment using two different n-gram language models: a class 2-gram and the class 4-gram used in the previous experiment.

Figure 5-9 summarizes our findings for the different language models. Even though we expected that the class 4-gram, with its longer distance constraints, might show a larger increase in errors due to new words, our experiment did not confirm this hypothesis. The class 4-gram shows better absolute word accuracy due to more powerful language constraints. These superior constraints help the in-vocabulary parts of the utterances more than they hurt due to contextual effects near new words. Even when we examine the error rates on the words adjacent to new words, we see virtually no difference on the number of indirect new-word errors between the 2-gram and 4-gram language models (not statistically significant at the 0.05 level). Part of the reason for this is that, as we explained in Section 5.3.2, in many cases the recognizer substitutes a word from the same word class as a new word. Because we are using *class*-conditional language models, within-class substitutions have no effect on the language model scores for the rest of the utterance.

5.3.4 Multiple New Words per Utterance

In the previous section we examined direct and indirect errors for utterances that contained only *one* new word per utterance. Do multiple new words per utterance cause more severe errors? New words could form sequences, or they could be disjoint. We address these two cases separately when examining the problems specific to multiple

[5] Due to the special new-word alignment used for the reduced-vocabulary system, insertions adjacent to new words were aligned with the new words and not the neighboring in-vocabulary words. Thus, there were zero insertions for adjacent words.

		sub		*ins*		*del*		*total*	
	n_{total}	n	%	n	%	n	%	n	%
2-gram									
baseline	322	20	6.3	8	2.5	4	1.3	32	10.0
reduced		70	21.7	17	5.3	0	0.0	87	27.0
Δ		+50	+15.5	+9	+2.8	−4	−1.3	+55	+17.1
4-gram									
baseline	322	9	2.8	10	3.1	2	0.6	21	6.5
reduced		54	16.8	21	6.5	0	0.0	75	23.3
Δ		+45	+14.0	+11	+3.4	−2	−0.6	+54	+16.8

Figure 5-9: Errors adjacent to new words versus language model. Errors measured on words adjacent to new words in $\mathcal{S}_1$. Although the absolute performance is better with the class 4-gram, there is very little difference in the degradation Δ between the class 2-gram and class 4-gram. The total number of reference words over which word errors were measured is indicated by n_{total}.

new words per utterance. We performed multiple-new-word experiments on test set $\mathcal{S}_{2+}$, which contains more than one new word per utterance.

Sequences of New Words

Sequences of new words are similar to single new words in that there is a contiguous region of an utterance containing out-of-vocabulary words. To the system, a sequence of new words is indistinguishable from a single new word, except possibly for duration differences. We might expect that a sequence of new words could cause additional indirect new-word errors. If a new-word sequence spans a longer region than a single new word, the recognizer may propose additional in-vocabulary words in its place. The greater number of words substituted in place of new words might cause additional language modeling problems in the vicinity of the new words since the language model context contains a greater number of direct new-word errors. Thus, we might expect increased indirect new-word errors due to *sequences* of new words.

We performed recognition experiments on the subset of $\mathcal{S}_{2+}$ in which all new words formed sequences (i.e., new words were not disjoint within an utterance). This subset, $\mathcal{S}_{2+/s}$, contained 101 utterances with a total of 202 new words.[6] Table 5-8 summarizes the direct new words associated with sequences of new words. Table 5-9 shows a sample of some of the reduced-vocabulary recognizer's hypotheses for sequences of new words.

[6] There were no sequences of three or more new words in our test set $\mathcal{S}$.

		sub		*ins*		*del*		*total*	
	n_{total}	*n*	%	*n*	%	*n*	%	*n*	%
baseline	202	9	4.5	2	1.0	0	0.0	11	5.4
reduced		159	78.7	43	21.3	36	17.8	238	117.8
Δ		+150	+74.3	+41	+20.3	+36	+17.8	+227	+112.4

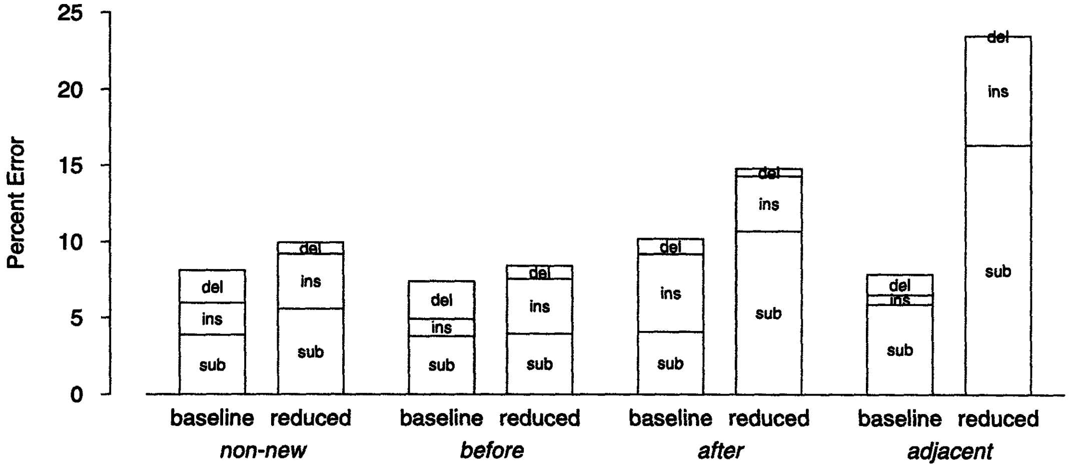
Table 5-8: Direct new-word errors associated with sequences of new words. The errors are measured for words aligned directly with the new words in $\mathcal{S}_{2+/s}$. Special new-word alignment was used for the reduced-vocabulary system's output. The total number of reference words over which word errors were measured is indicated by n_{total}. On average, there were 1.12 (227/202) direct errors per new word.

new words	→	*substitution*
Saint Petersburg	→	Pittsburgh
Saint Louis	→	Philly
Saint Louis	→	serve lunch
Los Angeles	→	Dallas
Los Angeles	→	Logan
Los Angeles	→	Boston
Las Vegas	→	what food is
Salt Lake	→	select
Westchester County	→	less first returning
Love Field	→	leaving
super saver	→	so December
Nevada Arizona	→	that goes on the

Table 5-9: Sample hypotheses for double new words. This table shows a randomly selected set of substitutions made by the reduced-vocabulary system for double new words, with the new words at the left and the hypothesized word(s) at the right.

While some of the substitutions contained more than one (in-vocabulary) word, many of them contained only a single word. As was the case with single new words (Table 5-7), some of the substitutions appeared to be motivated more by acoustics and others more by language constraints. In general, most of the double new words were not treated differently than single new words, at least in terms of direct new-word errors.

We also examined the indirect errors associated with the sequences of new words in $\mathcal{S}_{2+/s}$, summarized in Figure 5-10. As with single new words (Figure 5-8), the error rate of all non-new words did not increase very much, up only 1.9% (factor of 1.2), with a smaller increase than we saw with single new words. Words after sequences of new words were adversely affected more than words before, but not by a wide margin. Again, with sequences of new words, the most affected in-vocabulary words were those adjacent



(a)

	n_{total}	*sub*		*ins*		*del*		*total*	
		n	%	n	%	n	%	n	%
non-new									
baseline	803	31	3.9	17	2.1	17	2.1	65	8.1
reduced		45	5.6	29	3.6	6	0.7	80	10.0
Δ		+14	+1.7	+12	+1.5	−11	−1.4	+15	+1.9
before									
baseline	607	23	3.8	7	1.2	15	2.5	45	7.4
reduced		24	4.0	22	3.6	5	0.8	51	8.4
Δ		+1	+0.2	+15	+2.5	−10	−1.6	+6	+1.0
after									
baseline	196	8	4.1	10	5.1	2	1.0	20	10.2
reduced		21	10.7	7	3.6	1	0.5	29	14.8
Δ		+13	+6.6	−3	−1.5	−1	−0.5	+9	+4.6
adjacent									
baseline	153	9	5.9	1	0.7	2	1.3	12	7.9
reduced		25	16.3	11	7.2	0	0.0	36	23.5
Δ		+16	10.5	+10	+6.5	−2	−1.3	+24	*+15.7*

(b)

Figure 5-10: Indirect errors due to sequences of new words. The indirect errors are given for the test set $\mathcal{S}_{2+/s}$. Similar to the case of single new words, the largest degradation in performance is on the words adjacent to the new word sequences. Altogether, there were 202 new words in $\mathcal{S}_{2+/s}$, yielding an average of 0.07 (15/202) indirect errors per new word.

		sub		*ins*		*del*		*total*	
	n_{total}	n	%	n	%	n	%	n	%
baseline	1,218	36	3.0	7	0.6	13	1.1	56	4.6
reduced		1,052	86.4	162	13.3	309	25.4	1,523	125.0
Δ		+1,016	+83.4	+155	+12.7	+296	+24.3	+1,467	+120.4

Table 5-10: Direct new-word errors associated with disjoint new words. The errors are measured for words aligned directly with the new words in $\mathcal{S}_{2+/\text{d}}$. Special new-word alignment was used for the reduced-vocabulary system's output. The total number of reference words over which word errors were measured is indicated by n_{total}. On average, there were 1.20 (1,467/1,218) direct errors per new word.

to new words. The error rate on adjacent new words increased by 15.7% (factor of 3). Thus, sequences of new words have approximately the same effect on performance as single new words. The added length of out-of-vocabulary regions due to sequences of new words does not appear to cause a significant increase in indirect new word errors compared to single new words.

Disjoint New Words

While sequences of new words had an effect on recognizer performance similar to that of single new words, we might expect *disjoint* new words to have a more pronounced effect. Disjoint new words are out-of-vocabulary words that have in-vocabulary words between them. With more than one region of an utterance containing out-of-vocabulary words, the potential for misrecognition is increased.

We performed recognition experiments on the subset of $\mathcal{S}_{2+}$ in which there were at least two disjoint new words. This subset, $\mathcal{S}_{2+/\text{d}}$, contained 466 utterances with a total of 1,218 new words (2.6 new words per utterance on average).[7] Table 5-10 summarizes the direct errors associated with disjoint new words. Overall, the increase in word-error rate is similar to that of single new words. Similarly, the number of word errors per new word, 1.20, is similar as well. We conclude that as far as *direct* errors are concerned, disjoint new words are no worse than single new words.

Figure 5-11 summarizes our findings on indirect errors caused by disjoint new words.

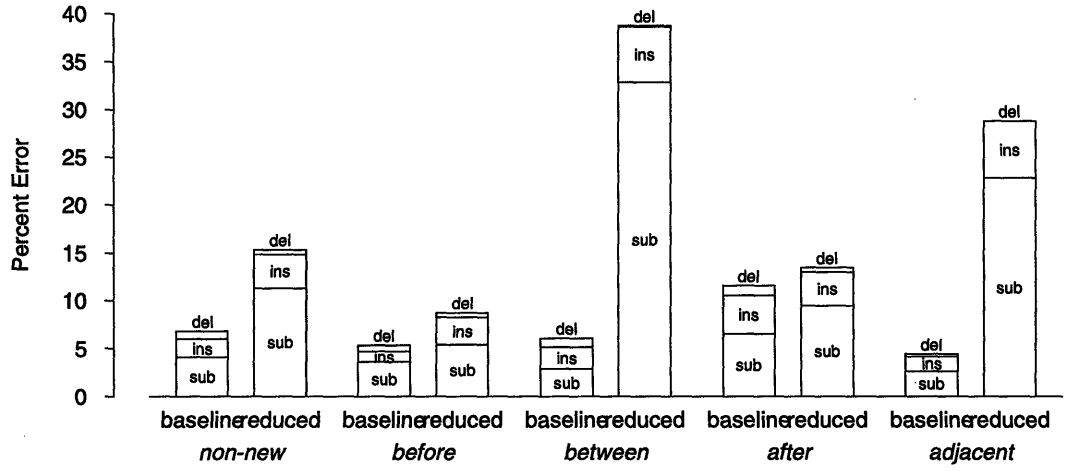
[7] We consider the utterance "Find me all the flights from *Milwaukee* to *St. Louis*," with three new words set off with italics, to belong to $\mathcal{S}_{2+/\text{d}}$ even though it does contain two new words in sequence. The defining characteristic of the disjoint set of utterances is that it contains at least on in-vocabulary word surrounded by new words.

Overall, we found that the error rate on all in-vocabulary words increased by 8.5% (factor of 2.3). This increase was significantly larger than that for either single or sequences of new words. We found that recognition of words adjacent to new words was again severely affected, with an increase in word-error rate of 24.4% (factor of 6.6). This performance degradation was again more severe than for single or consecutive new words. Examining the words *between* new words revealed where the additional errors due to disjoint new words occurred. Words between new words were misrecognized more than any other class of words; their error rate increased by 32.7% (factor of 6.4).

We suspected that the general form of utterances containing at least two disjoint new words may have been strongly influenced by the ATIS task and by our selection of simulated new words. Since over one third of ATIS utterances contain a phrase of the form "*city-name* to *city-name*" and most of our new words were city names, we would expect that many of our utterances containing disjoint new words would follow this form. In $\mathcal{S}_{2+/\mathrm{d}}$, we found that fully 71% of the utterances were of the form "*new-word* to *new-word*". Thus, our set of disjoint new words may not be representative of what we might see in other tasks. In our experiment, 54% of *between* words were the word "to." Since this is a short function word, we might hypothesize that part of the reason *between* words perform so poorly is that they are dominated by the word "to," which may be poorly recognized by the system regardless of the presence of new words. However, we examined the baseline system's performance on the word "to" and found that the word-error rate on "to" was only 3.2% over all $\mathcal{S}$, compared to the overall word-error rate of 8.1%. In other words, "to" is recognized *much better* than the average word by the baseline system. Therefore, we concluded that recognition of "to" is severely affected when it is surrounded by new words. Because it is a short word, it seems that the system often "absorbs" the word into its hypotheses for the nearby new words.

5.3.5 Summary of Recognition Errors Caused by New Words

In summary, we performed carefully controlled experiments in which we used a baseline system that encountered no new words and a reduced-vocabulary system that did so that we could compare system performance on the same set of utterances. This allowed



(a)

	n_{total}	*sub* n	*sub* %	*ins* n	*ins* %	*del* n	*del* %	*total* n	*total* %
non-new									
baseline	4,009	163	4.1	77	1.9	32	0.8	272	6.8
reduced		454	11.3	142	3.5	17	0.4	613	15.3
Δ		+291	+7.3	+65	+1.6	−15	−0.4	+341	+8.5
before									
baseline	2,429	87	3.6	26	1.1	17	0.7	130	5.4
reduced		131	5.4	70	2.9	12	0.5	213	8.8
Δ		+44	+1.8	+44	+1.8	−5	−0.2	+83	+3.4
between									
baseline	740	21	2.8	17	2.3	7	0.9	45	6.1
reduced		243	32.8	43	5.8	1	0.1	287	38.8
Δ		+222	+30.0	+26	+3.5	−6	−0.8	+242	*+32.7*
after									
baseline	840	55	6.5	34	4.0	8	1.0	97	11.5
reduced		80	9.5	29	3.5	4	0.5	113	13.5
Δ		+25	+3.0	−5	−0.6	−4	−0.5	+16	+1.9
adjacent									
baseline	1,303	34	2.6	20	1.5	3	0.2	57	4.4
reduced		297	22.8	78	6.0	0	0.0	375	28.8
Δ		+263	+20.2	+58	+4.5	−3	−0.2	+318	*+24.4*

(b)

Figure 5-11: Indirect errors due to disjoint new words. The indirect errors are given for the test set $\mathcal{S}_{2+/\text{d}}$. The largest performance degradations were for in-vocabulary words between new words, followed by in-vocabulary words adjacent to new words. Altogether there were 1,218 new words in $\mathcal{S}_{2+/\text{d}}$, yielding an average of 0.20 (242/1,218) indirect errors per new word.

		Δ%
$\mathcal{S}$	non-new	+3.8
	before	+0.3
	between	+8.7
	after	−0.2
	adjacent	*+22.2*
$\mathcal{S}_1$	non-new	+3.8
	before	+4.1
	after	+3.3
	adjacent	*+16.8*
$\mathcal{S}_{2+/\mathrm{s}}$	non-new	+1.9
(sequence)	before	+1.0
	after	+4.6
	adjacent	*+15.7*
$\mathcal{S}_{2+/\mathrm{d}}$	non-new	+8.5
(disjoint)	before	+3.4
	between	*+32.7*
	after	+1.9
	adjacent	*+24.4*

Table 5-11: Summary of indirect new-word errors. Δ% indicates the difference between the reduced-vocabulary and baseline systems' total word-error rates.

us to analyze the errors that were due *solely* to the introduction of simulated new words by the reduction of the baseline vocabulary. Over all evaluation utterances in the full evaluation set $\mathcal{S}$, we found that there were 1,618 new words. Comparing the performance of the reduced-vocabulary and baseline systems, we found a total of 2,368 errors (out of 12,707 reference words) attributable solely to the reduction in vocabulary. The overall word-error rate on the $\mathcal{S}$ utterances was 26.7% versus 8.1% for the baseline system, a significant increase. On average, the reduced vocabulary system experienced an increase of 1.46 errors per new word. Breaking this down into direct and indirect errors, we found 1.20 (1,936/1,618) direct and 0.27 (432/1,618) indirect errors per new word.

Table 5-11 summarizes the increase in word-error rate on the in-vocabulary words (i.e., indirect errors). We found that even one new word could have a significant negative effect on recognition, with performance on in-vocabulary words *adjacent* to it being the most affected. Overall, we found no significant difference in performance degradation for words before and after new words.

We found that sequences of new words caused indirect errors at similar levels to isolated new words. Therefore, new word sequences can be thought of essentially as single new words, perhaps with longer durations. When we examined utterances containing disjoint new words, utterances in which there were in-vocabulary words surrounded by new words, we found that the number of indirect new word errors increased substantially. Although words adjacent to new words were once again adversely affected, words *between* new words suffered the greatest increase in performance degradation.

5.4 Search Complexity and New Words

In Section 5.3.1 we examined recognition errors caused by the occurrence of new, out-of-vocabulary words. However, not only do new words increase the number of system errors (e.g., word-error rate), they also increase the computation complexity of the recognition search. The primary reason for both increases is that the system (without explicit new-word modeling/detection capabilities) is forced to model the acoustics of new words with arbitrary combinations of in-vocabulary words. Typically, there are a large number of in-vocabulary words "competing" to fill the region where a new word occurs, none of them being a clear winner.

In Section 4.4.2 we introduced diagnostic tools based on our word graph representation. We explained that the word graphs we compute represent a recognition "history," in that they contain all word hypotheses (word labels, scores, and time alignments) that are explored during our A* N-best search or our A* word graph search. Because they show the history of the recognition search, we can examine the computational demands of the search empirically by examining word graphs.

One of the tools we introduced in Chapter 4 was the *word lattice* display. Figure 5-12 shows a word lattice for an utterance containing a new word, and Figure 5-13 shows an utterance that is entirely in-vocabulary. (See Section 4.4.2 for details on the computation of word lattice displays.) The word lattices show word hypotheses, their time spans, and the combination of their acoustic and lexical scores. The vertical axis is acoustic/lexical score, and the horizontal axis is time. By comparing Figure 5-12 and Figure 5-13, it is

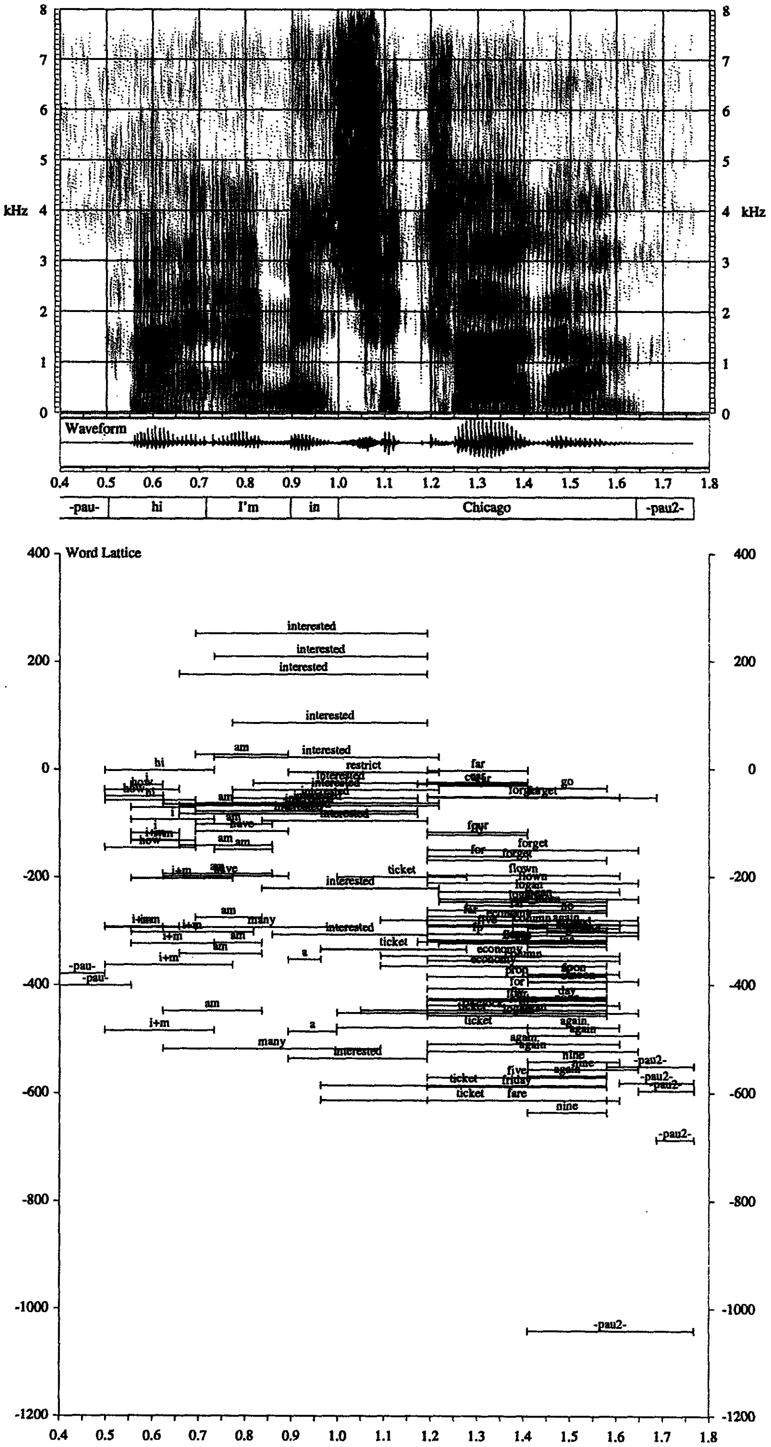


Figure 5-12: Word lattice for utterance containing an out-of-vocabulary word. In this example, the word "Chicago" is a new word and is responsible for a large number of word competitors. In the word lattice, the vertical axis represents the acoustic/lexical score, and the horizontal axis represents time.

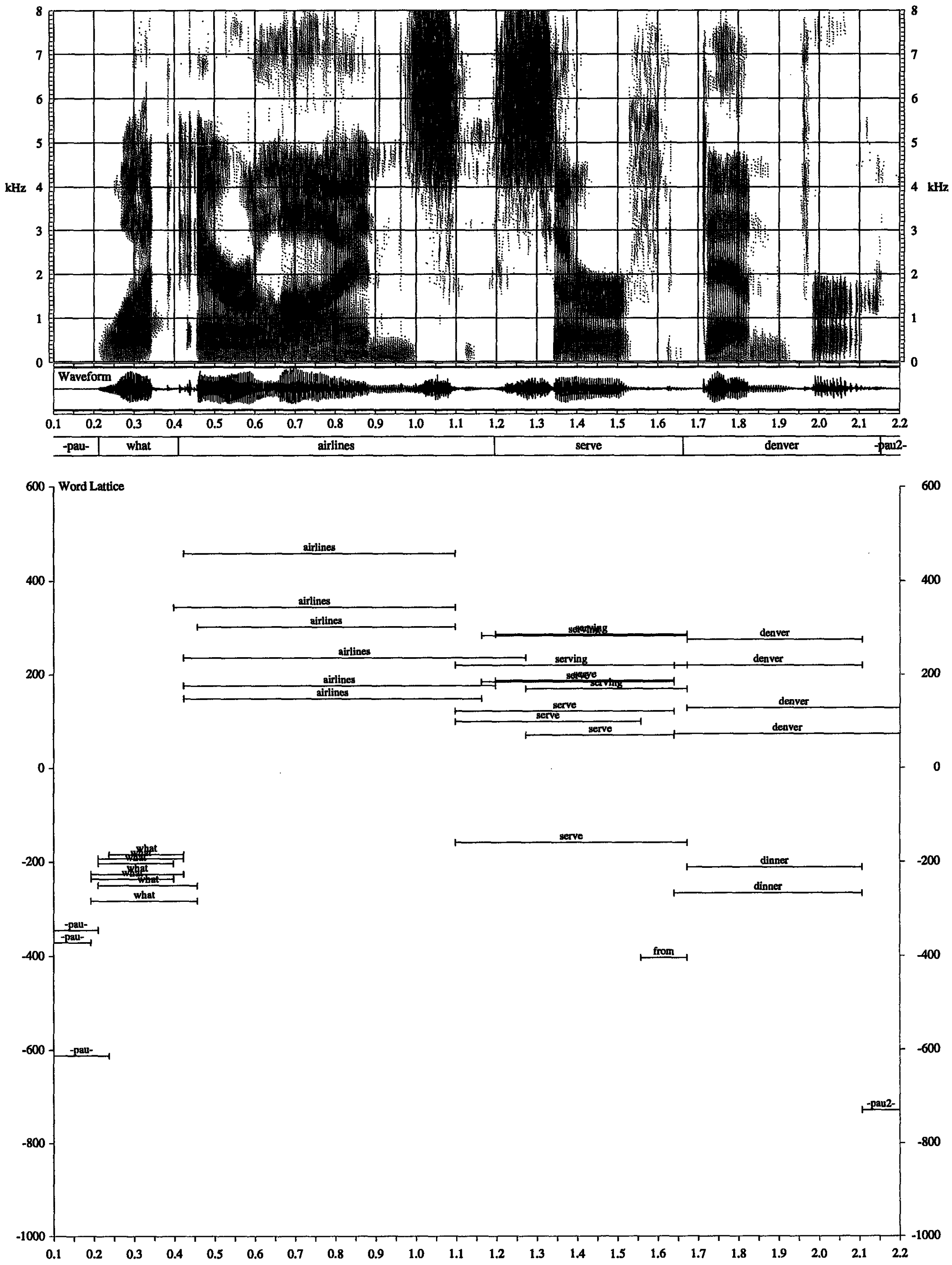


Figure 5-13: Word lattice for an utterance containing only in-vocabulary words. In the word lattice, the vertical axis represents acoustic/lexical score, and the horizontal axis represents time.

	total edges	*edges/second*
$\mathcal{S}_0$	1,248	390
$\mathcal{S}_1$	5,969	1,441
$\mathcal{S}_{2+}$	12,671	2,374

Table 5-12: Word graph complexity by test set. The second column displays the geometric mean for the number of total word graph edges. The third column displays the geometric mean for the normalized number of word graph edges (edges/second).

clear that new words can greatly increase the number of word hypotheses, particularly in regions near them. For the new word "Chicago," there are over 30 in-vocabulary words competing with one another at the specified relative score threshold $\theta = 300$. (The recognizer's first-choice hypothesis for the words "in Chicago" was "interested four.")

5.4.1 Overall Increase in Computation Due to New Words

In order to gauge the overall increase in computation due to the occurrence of new words, we measured statistics on the word graphs of all utterances in our full evaluation set $\mathcal{S}$. Because the time required to compute word graphs is approximately proportional to the number of word edges contained in them, we used this number as a measure for computation time.

Table 5-12 displays the geometric mean[8] number of edges per utterance and number of edges per second for the evaluation sets $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_{2+}$. Because the average utterance length increased over $\mathcal{S}_0$ to $\mathcal{S}_1$ to $\mathcal{S}_{2+}$, we felt that the edge counts normalized by utterance length (i.e., edges per second) were more meaningful for comparison purposes. Clearly, utterances that contain new words result in larger, more computationally expensive word graphs. This indicates that computation time increases substantially when new words are present. In our experiment, utterances in $\mathcal{S}_1$ required more than three times the computational effort per second as utterances in $\mathcal{S}_0$. This increase is due to the increased number of word hypotheses competing with one another in the recognition search, which in turn is due to the recognizer's difficulty in modeling the acoustics of new words using in-vocabulary words.

[8] We chose the geometric mean since the number of word edges has a distribution that is nearly log-normal.

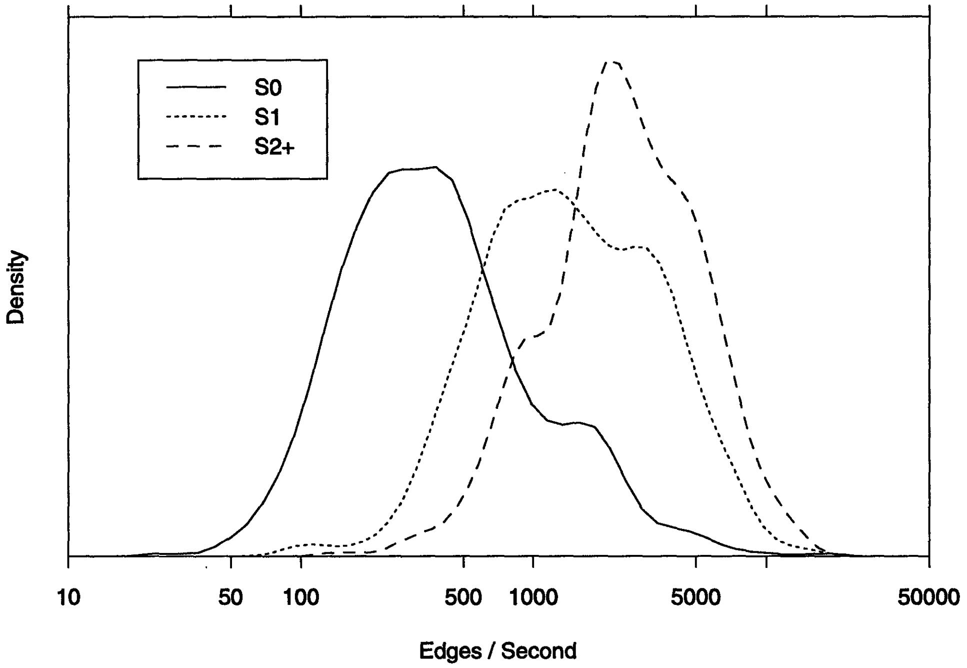


Figure 5-14: Distribution of number of word graph edges per second by test set.

Because there is a large utterance-to-utterance variation in the computational requirements, we also examined the *distribution* of number of edges per second. Figure 5-14 shows the estimated probability densities[9] for number of edges per second for the sets $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$. Utterances in $\mathcal{S}_0$, which contain no new words, are computationally far less demanding the those in $\mathcal{S}_1$ and $\mathcal{S}_2$. Not only is the edge rate low for $\mathcal{S}_0$, the variance on the edge rate is also relatively low, as compared to the means and variances for $\mathcal{S}_1$ and $\mathcal{S}_{2+}$ utterances. In general, the more new words per utterance, the greater the mean and variance on the amount of computation.

5.4.2 Dependence on Position of New Words

We demonstrated that new words are responsible for increased computational complexity during the recognition search. The more new words there are in an utterance, particularly when they are disjoint, the more word hypotheses that the search has to consider. We were curious to see if the amount of computation was dependent on the

[9] The densities were estimated non-parametrically using a Gaussian smoothing window.

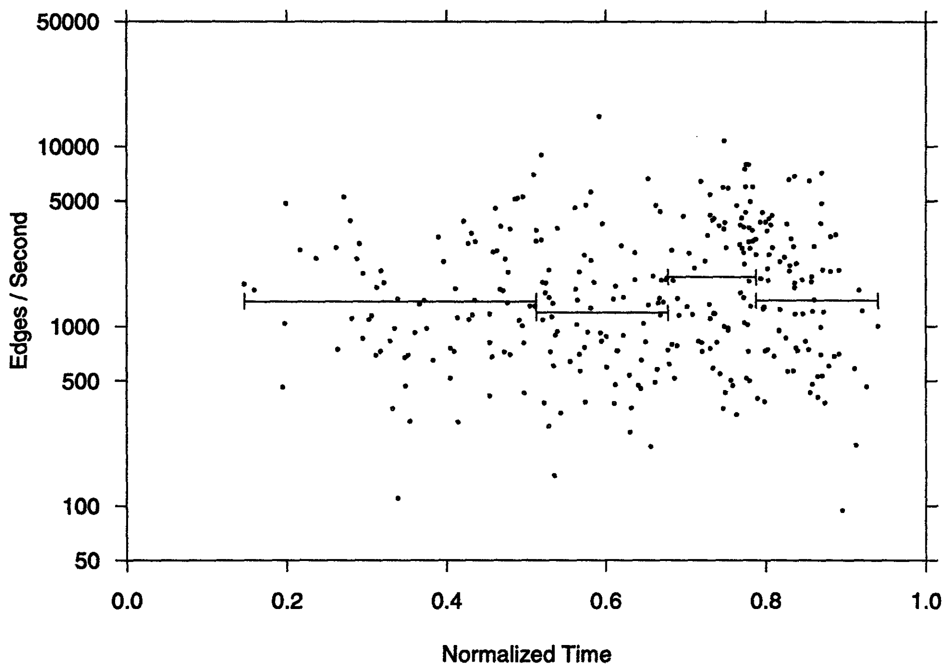


Figure 5-15: Dependence of computational complexity on new-word position. This plot shows the scatter plot of word graph edges per second versus new-word position for utterances in $\mathcal{S}_1$. The position of a new word is represented by a single normalized time value $t' = (t_1 + t_2)/2d$, where t_1 and t_2 are the endpoints of the word and d is the duration of the utterance. This value t' represents the normalized time of the midpoint of the new word. The four horizontal lines indicate the geometric mean number of edges per second for each of the four quartiles based on t'.

position of new words within utterances. Do new words near the beginning of utterances cause more or less computational complexity compared to new words near the end of utterances? Depending on the direction and organization of the search, we might hypothesize that the position of new words has an effect on computation time.

We performed an experiment to study the dependence of computation on new-word position. Again, we used number of word graph edges per second as our measure of computation time. Figure 5-15 shows, as a scatter plot, the number of edges per second versus new-word position for the set of utterances containing exactly one new word per utterance, $\mathcal{S}_1$. New-word position is represented by the time midpoint, normalized by utterance duration. We have overlayed the geometric mean number of edges per second for four separate regions of the utterances. The four regions were determined by the quartiles of normalized new-word position. Overall, there appears to be very little, if any, dependence of computation time on new-word position. New words at the beginning of utterances require approximately the same amount of computation as new words at the middle or end of utterances.

This lack of dependence on new-word position should not be surprising. The fact that we are using word graphs means that variability of recognizer hypotheses (e.g., due to a new word) at all points of the search can be represented efficiently. Since the recognition search space is not unfolded into a tree in the word graph computation, variability near the beginning of the search is no worse than variability near the end of the search. In contrast, if the search space were unfolded into a tree (e.g., with the A* N-best search, see Section 4.2) then variability near the root of (the search tree would result in duplicate path extensions throughout the rest of the search. In the case of a search tree, variability near the root is worse than variability near the leaves. Theoretically and empirically, the position of new words within an utterance has little effect on the computational complexity of generating a *word graph.*

5.4.3 Active-Word Counts and their Correlation with New Words

In Section 4.4.2 we presented a novel measure of word hypothesis competition during search called the *active-word count.* This count can be computed for each time slice within a word graph and represents the number of distinct words from the system's vocabulary that are *active* at that particular time. By active, we mean that the words are competitive during the search because their score is above a specified relative score threshold θ.

Figure 5-16 shows the active-word counts for an utterance containing an out-of-vocabulary word. As described in Section 4.4.2, the active word count is computed for each time slice by counting the number of distinct words associated with word graph edges that cross it. The figure shows that the number of distinct words active in the search tends to be higher in the vicinity of new words. Examining many such plots, we observed that the active-word counts tended to be higher in utterances containing new words. In addition, peaks or rapid increases in the number of active words tend to occur near new words. Based on these observations, we hypothesized that the active-

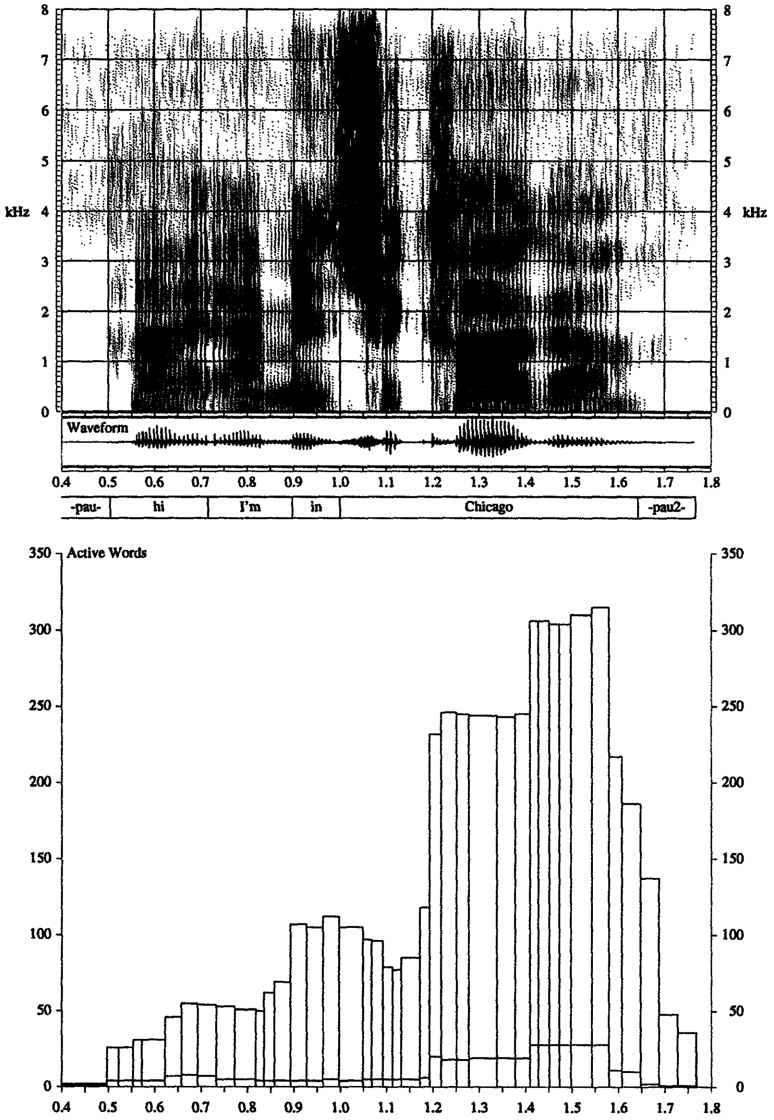


Figure 5-16: Active-word counts for an utterance containing a new word. In this example, the word "Chicago" is a new word and is responsible for a large number of word competitors. The active-word count is plotted for two different relative score thresholds θ (300 and 800).

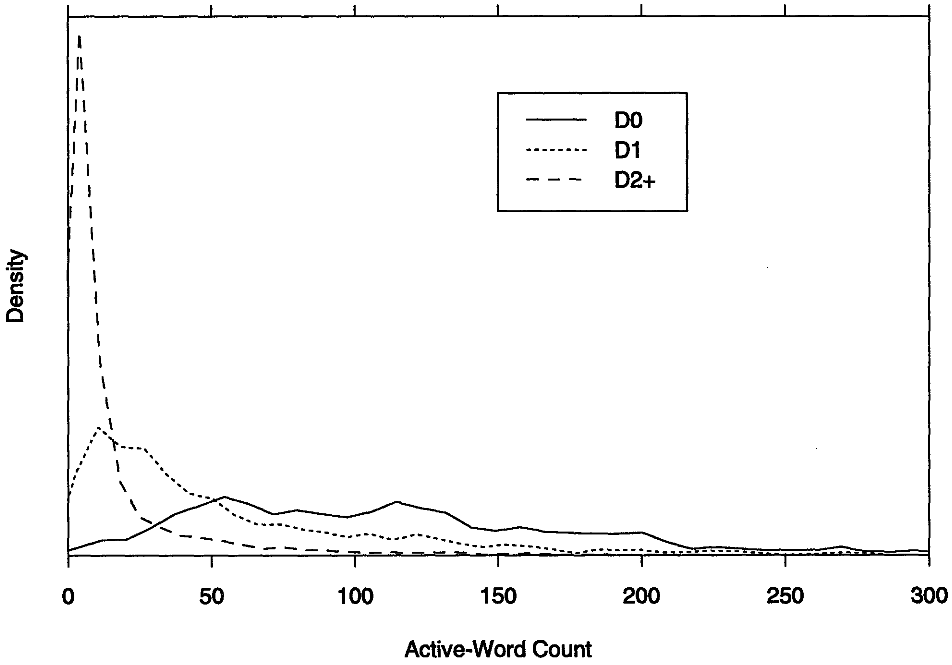


Figure 5-17: Distribution of active-word counts versus new-word distance. The density of the active-word count is given for three sets of words in $\mathcal{S}_1$: new words ($\mathcal{D}_0$), words adjacent to new words ($\mathcal{D}_1$), and words with at least one word between them and the nearest new word ($\mathcal{D}_{2+}$).

word count might be a useful feature for new-word detection.

We performed a simple experiment to examine the correlation between high active-word counts and the occurrence of new words without actually detecting new words. For each reference word in $\mathcal{S}_1$, we computed the time-average active-word count and the distance, measured in number of reference words, from the nearest new word. We then divided the words into three groups: $\mathcal{D}_0$, those with new-word distance zero (i.e., new words); $\mathcal{D}_1$, those with distance one; and $\mathcal{D}_{2+}$, those with distance greater than or equal to two. We examined the distribution of the active-word count for each group to see if it was dependent on proximity to new words. Figure 5-17 shows the active-word distributions for each of the three word groups. It shows that the active-word counts associated with new words ($\mathcal{D}_0$) tend to be the highest, followed by the counts associated with adjacent words ($\mathcal{D}_1$). Words a distance of at least two words from new words ($\mathcal{D}_{2+}$) show the lowest active-word counts. The means are 113, 55, and 15 for $\mathcal{D}_0$, $\mathcal{D}_1$, and $\mathcal{D}_{2+}$, respectively. The variances on these counts are relatively large as evident in the distributions shown in Figure 5-17.

Although new-word detection per se is beyond the scope of this thesis, we have demonstrated that the active-word count may be a useful feature for detection because it is correlated with the *location* of new words. We have observed that recognizer "confusion," as measured by the number of word edges or number of active words, increases in the vicinity of out-of-vocabulary words. Measuring this confusion could aid in detecting new words. We leave this for future work.

5.5 Learning New Words: Training Issues

In Sections 5.3 and 5.4 we examined the effect of new words on recognition in terms of performance degradation and computational complexity. Combined with the lexical, phonological, and linguistic study of Chapter 2 we hope these results will lead to improved methods for detecting new words.

Even after new words are detected, either automatically or through user intervention, we are still faced with the problem of incorporating new words into a system's vocabulary, or *learning* them. After all, a user may want to continue to use a new word and have the system treat it as if it were a part of its regular vocabulary. Adding a word to a system's vocabulary may involve the updating of several major recognition and understanding components because the system's knowledge of a word (e.g., how it is pronounced, how it can be used within the language, and what it means) is spread throughout a system's models. Many or all of the various system models may need updating in order to incorporate new words.

In this section we examine issues related to learning or incorporating new words within the SUMMIT continuous-speech recognition system. The major system components that are most likely to require updating when learning new words are:

1. the acoustic models,
2. the lexical (pronunciation) models,
3. the language models, and

4. the language models (e.g., grammar and meaning representations) associated with the natural language component.

In Section 5.5.1, we examine the importance (or lack thereof) of updating context-*independent* acoustic models. In Section 5.5.2, we examine the creation of lexical models for new words within the SUMMIT system without the need for training data containing the new words. Finally, in Section 5.5.3 we examine the process of adding new words to the language models used for recognition. We do not examine issues related to adding new words to a natural language component in this thesis. Overall, the primary goal is to determine characteristics of the above models that are helpful for new-word incorporation and to quantify how updating these models affects performance, both on words in the original system vocabulary and on newly learned words.

Our research was conducted within the context of a single system, the SUMMIT system, so that we could make controlled comparisons. Because we already had a system, the baseline system, that was fully trained on our simulated new words, we could use its performance on the new words as an upper bound on how well our particular new words could be learned. In our experiments, we examined the three sets of models, the acoustic models, the lexical models, and the n-gram language models, separately. The goals were to determine for which components retraining was most important and to determine features of the various components that aid in the incorporation of new words.

5.5.1 Acoustic Models

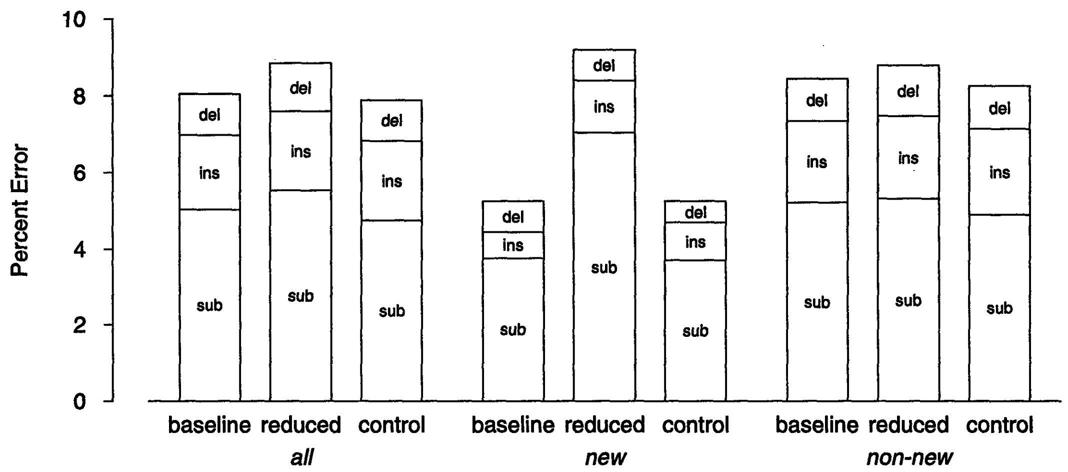
One reason for using sub-word units in acoustic modeling is that the units are shared among words. If the set of units is complete (e.g., all phonemes are represented) we should, in principle, be able to model any new word using them. The most common sub-word units used in current speech recognition systems are context-independent and context-dependent phonetic units.

In our experiments, we used only the context-*independent* phonetic models of the SUMMIT system. We would expect that context-independent phonetic models enable easy new-word learning. Because the models are context-independent, they are shared by a large number of words in the system vocabulary. Because of this sharing, we would expect that such models would not be greatly affected (improved) by the addition of a relatively small amount of training data for new words to be learned. Therefore, we expected that training on examples of new words would not improve performance significantly.

To test this hypothesis, we conducted a controlled experiment using three systems: a baseline system, a test system, and a control system. The three systems differed only in their acoustic models. The baseline system's acoustic models were trained fully on the new words and was used as a yardstick by which to measure the shortfall in performance of the test system. The test system's acoustic models were trained on utterances that did not contain any of the new words. This selection of utterances based on vocabulary reduced the training set by about 17%. The control system's acoustic models were trained on utterances that contained the new words, but the size of the training set was comparable to that of the test system. (See Table 5-4 for the sizes of the three training sets.) All systems shared the same large vocabulary, lexical models, and language model (class 4-gram).

Figure 5-18 summarizes the recognition performance of the three systems over the full test set $\mathcal{S}$. The performances were measured three ways: over all words, over all new words (i.e., the words being learned), and over all non-new words. Overall, the word-error rates for the three systems were comparable, averaging 8.3%. However, when we examined the performance on only the new words being learned, we found that the error rate for the test system was 4% worse (factor of 1.8 times higher) than for the baseline system. Training the context-independent acoustic models with examples of the new words (in the baseline system) reduce the error rate. Thinking that the better performance of the baseline system could have been due to the larger training set size, we compared the performance of the test system to that of the control system, which was trained on approximately the same number of utterances as the test system. However, the performance of the control system was virtually identical to that of the baseline system, both overall and on new words only.

We were disturbed by the result that lack of new word training data could have such



(a)

	n_{total}	*sub* n	*sub* %	*ins* n	*ins* %	*del* n	*del* %	*total* n	*total* %
all									
baseline	12,707	640	5.0	247	1.9	136	1.1	1,023	8.1
reduced		704	5.5	261	2.1	160	1.3	1,125	8.9
Δ		+64	+0.5	+14	+0.1	+36	+0.3	+102	+0.8
control		603	4.8	265	2.1	134	1.1	1,002	7.9
Δ		−37	−0.3	+18	+0.1	−2	−0.0	−21	−0.2
new									
baseline	1,618	61	3.8	11	0.7	13	0.8	85	5.3
reduced		114	7.1	22	1.4	13	0.8	149	9.2
Δ		+53	+3.3	+11	+0.7	+0	+0.0	+64	+4.0
control		60	3.7	16	1.0	9	0.6	85	5.3
Δ		−1	−0.1	+5	+0.3	−4	−0.2	+0	+0.0
non-new									
baseline	11,089	579	5.2	236	2.1	123	1.1	938	8.5
reduced		590	5.3	239	2.2	147	1.3	976	8.8
Δ		+11	+0.1	+3	+0.0	+24	+0.2	+38	+0.3
control		543	4.9	249	2.3	125	1.1	917	8.3
Δ		−36	−0.3	+13	+0.1	+2	+0.0	−21	−0.2

(b)

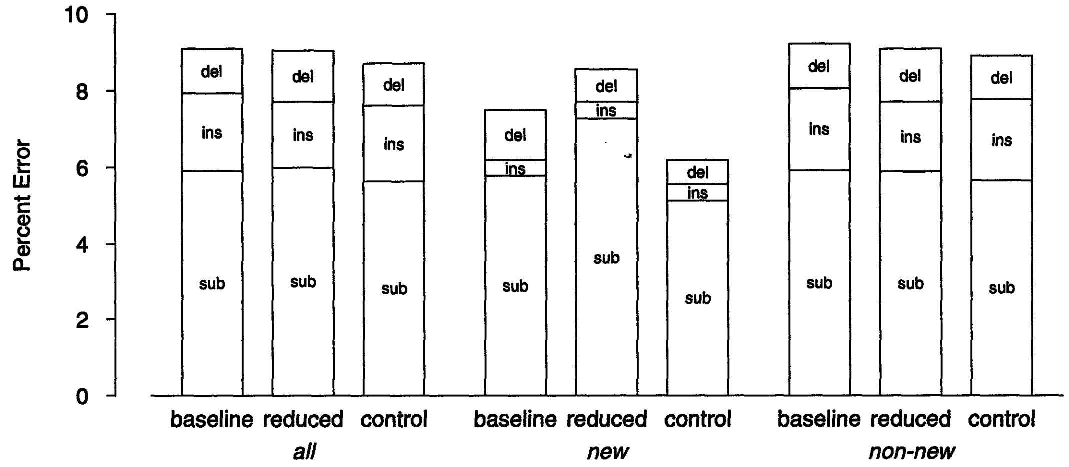
Figure 5-18: Importance of updating acoustic models. The increase in errors due to the reduced-vocabulary acoustic models as compared to the fully trained baseline acoustic models. The reduced-vocabulary models were not updated in any way.

a large effect on context-*independent* acoustic models. We thought that such models would be relatively insensitive to the vocabulary in use. Therefore, we examined the training of the baseline and reduced-vocabulary models in detail and discovered the reason for the large performance difference: a few of the models were very poorly trained due to sparse data in the reduced-vocabulary system, and these models were required in order to model some of the new words. Specifically, for the female model of [o^{y}], there were *zero* examples in the reduced-vocabulary training set. For the male models of [o^{y}] and [n̩], the reduced-vocabulary training set contained 80% fewer examples compared to the baseline-vocabulary training set.[10] These differences in number of training tokens are very large and unexpected for context-independent acoustic modeling.

In order to eliminate the effects of these particularly poorly trained models, we re-evaluated the baseline, reduced-vocabulary, and baseline-vocabulary control systems on the subset of the $\mathcal{S}$ utterances that did not contain new words requiring these models. All together we discarded a total of 588 utterances from $\mathcal{S}$ that required [o^{y}] or the male [n̩], yielding 861 utterances for evaluation. Figure 5-19 summarizes the performances of the three systems evaluated on the reduced test set. Clearly, the performance difference between the baseline acoustic models and the reduced-vocabulary acoustic models is significantly reduced, leading us to believe that the [o^{y}] and [n̩] models had a significant effect on the degradation that was evident in Figure 5-18.

This failure of the reduced-vocabulary models brings up an interesting point. While we might expect a set of context-*independent* models to be relatively vocabulary-independent, this expectation depends on having a set of *adequately* trained models to begin with. In our case, the reduced-vocabulary acoustic models were trained on utterances with a vocabulary of at most 1,326 words. Even with a vocabulary of this size and nearly 17,000 training utterances, it is possible for some of the rare phonetic units to be inadequately trained. Clearly, if we intend to build a system that is capable of incorporating new words we must ensure that all relevant context-independent acoustic models are properly trained. One way we could accomplish this would be to supplement

[10] [o^{y}] is the vowel in "boy," and [n̩] is a syllabic [n] that might occur at the end of "cotton" (i.e., [hæp$^{\square}$pn̩] instead of [hæp$^{\square}$pən]).



(a)

		sub		*ins*		*del*		*total*	
	n_{total}	n	%	n	%	n	%	n	%
all									
baseline	6,512	386	5.9	131	2.0	76	1.2	593	9.1
reduced		391	6.0	111	1.7	88	1.4	590	9.1
Δ		+5	+0.1	−20	−0.3	+12	+0.2	−3	+0.0
control		367	5.6	129	2.0	72	1.1	568	8.7
Δ		−19	−0.3	−2	−0.0	−4	−0.1	−25	−0.4
new									
baseline	467	27	5.8	2	0.4	6	1.3	35	7.5
reduced		34	7.3	2	0.4	4	0.9	40	8.6
Δ		+7	+1.5	+0	+0.0	−2	−0.4	+5	+1.1
control		24	5.1	2	0.4	3	0.6	29	6.2
Δ		−3	−0.6	+0	+0.0	−3	−0.6	−6	−1.3
non-new									
baseline	6,045	359	5.9	129	2.1	70	1.2	558	9.2
reduced		357	5.9	109	1.8	84	1.4	550	9.1
Δ		−2	−0.0	−20	−0.3	−6	−0.1	−8	−0.1
control		343	5.7	127	2.1	69	1.1	539	8.9
Δ		−16	−0.3	−2	−0.0	−1	−0.0	−19	−0.3

(b)

Figure 5-19: Updating acoustic models (revised). Utterances in $\mathcal{S}$ requiring [oʸ] or male [ŋ̩] context-independent acoustic models were discarded.

the task-specific training data with some general, non-task-specific data such as from the TIMIT utterances. In our case, we could have added training tokens of [o^y] and [ŋ̩] to train these models when the task-specific training data was insufficient for these models. Therefore, taking some minor precautions when training a system's acoustic models could improve its ability to incorporate new words.

Because we did not perform any experiments with context-dependent acoustic models, we can only speculate about how important retraining them is during new-word learning. Context-dependent modeling is known to be prone to sparse data problems. Often there is not enough training data available to adequately model all distinct (phonetic) contexts. Furthermore, the set of context-dependent models of a system may not include all those needed for previously unseen new words. If there are missing context-dependent models, or the models are not adequately trained, training on new-word examples could improve performance. Without actually performing experiments with context-dependent models, we cannot quantify the importance of updating context-dependent acoustic models when learning new words.

5.5.2 Lexical Models

When adding new words, lexical or pronunciation models for them *must* be added to the system. Depending on how the recognition system models word pronunciation, adding the pronunciation for a new word could be as simple as entering a base-form pronunciation (e.g., phonemic string) or as complex as constructing a pronunciation graph with transition probabilities or weights. The SUMMIT system falls into the latter category, but the complicated pronunciation models for new words *can* be added without any training on examples of the new words.

In general, a base-form pronunciation for a new word usually is needed to construct pronunciation models and can be generated in several ways including:

- by hand;

- by dictionary lookup;

- by performing phonetic recognition on utterances of the new word;[11]
- by using a text-to-speech system; or
- by some combination of the above (e.g., the work of Asadi et al. [3–5]).

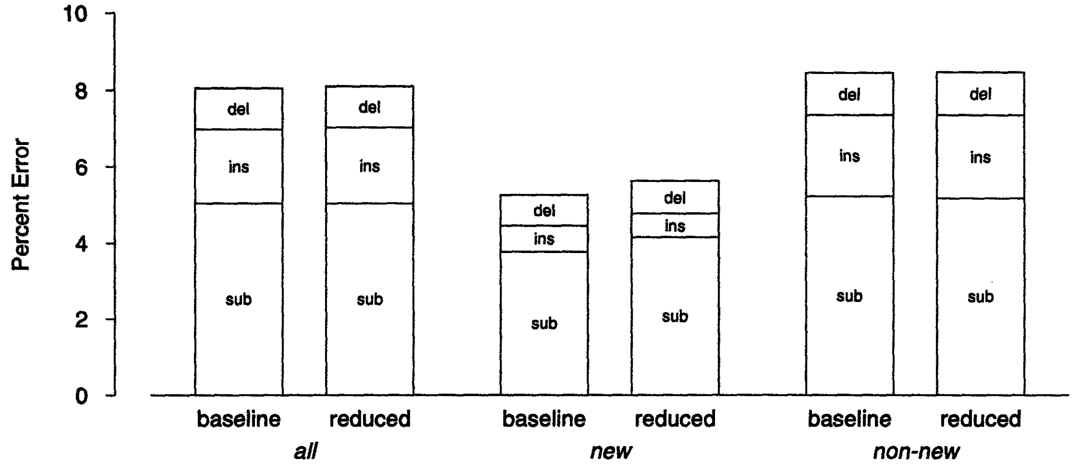
In the SUMMIT system, we start with a phonemic base-form, which could be entered by hand or obtained from a dictionary. Then, a set of phonological rules is applied which transforms the phonemic string into a phonetic graph. In the corrective training stage of SUMMIT, the arc weights on the phonetic graph are updated, requiring samples of the word in the training set. We evaluated how well the SUMMIT system could incorporate lexical models of new words without *any* examples of them for training. The only parts of SUMMIT's lexical models that require training data are the arc weights. If we set the arc weights for new words to be zero (or any suitable constant), we do not require any utterances of new words.

We used two systems for our experiment: a baseline system and a test system, that differed only in their lexical models. The baseline system was the system fully trained on the new words. The test system was the same as the baseline system except that the lexical arc weights associated with the new words had been set to zero. Thus, the lexical models of the test system could have been derived in the absence of new word examples.

Figure 5-20 summarizes the performance results of the baseline and test systems on the full test set $\mathcal{S}$. Overall performance was virtually identical between the two systems. Over the new words only, the word-error rate increased by only 0.4% (factor of 1.1). Evidently, the lexical arc weights were unimportant for the new words. Training the lexical models on examples of the new words, as in the baseline system, did not improve performance significantly. These results suggest that we do not need to worry about computationally expensive corrective training when adding new words.

The insignificance of the lexical arc weights during the incorporation of new words surprised us. Overall, the addition of arc weights in general to the SUMMIT system has

[11] Using phonetic recognition to derive a pronunciation obviously requires spoken utterances of the new word to be collected.



(a)

	n_{total}	*sub*		*ins*		*del*		*total*	
		n	%	*n*	%	*n*	%	*n*	%
all									
baseline	12,707	640	5.0	247	1.9	136	1.1	1,023	8.1
reduced		640	5.0	252	2.0	138	1.1	1,030	8.1
Δ		+0	+0.0	+5	+0.0	+2	+0.0	+7	+0.1
new									
baseline	1,618	61	3.8	11	0.7	13	0.8	85	5.3
reduced		67	4.1	10	0.6	14	0.9	91	5.6
Δ		+6	+0.4	−1	−0.1	+1	+0.1	+6	+0.4
non-new									
baseline	11,089	579	5.2	236	2.1	123	1.1	938	8.5
reduced		573	5.2	242	2.2	124	1.1	939	8.5
Δ		−6	−0.1	+6	+0.1	+1	+0.0	+1	+0.0

(b)

Figure 5-20: Updating lexical models.

resulted in significant improvement in performance [75]. However, there is a possible explanation for why we found very little improvement with the arc weights for new words. Words that were frequent (e.g., "to," "from," "flights," and "the," the most frequent words in ATIS) were more likely to benefit from the corrective training of the arc weights. These words occurred more often in the training set and thus were more involved in the corrective training algorithm. Relatively infrequent words, such as our simulated new words, did not enter into the corrective training process as often because of their low frequency. Thus, the lexical arc weights of infrequently occurring words may not deviate from zero as much as those in more common words.

5.5.3 Language Models

Language models are a critical part of speech recognition systems because they provide strong constraints on word sequences. In order to add a new word to a system, the language model must be updated in a way that allows the word to be a part of allowable word sequences. Not only must the new word be enabled by a language model, but the probabilities involving the new word must be high enough that the word is not penalized too severely with respect to others in the vocabulary, because the language model's probabilities can have a large impact on recognizer scores. If a language model is not updated appropriately when a new word is added, that word may not ever be included in the recognizer's top choice.

In general, n-gram language models contain a great number of probability estimates, particularly for $n > 2$. If the language model is a *word* n-gram, where the probability for a given word depends on the identity of the preceding $n - 1$ words, the number of probabilities can be very large indeed, requiring an enormous set of training text to estimate them. If we want to add a new word to such a model, we are faced with a problem: how do we estimate the needed probabilities associated with the new word if we have little or no text containing the new word. Jelinek et al. [34] presented a method requiring a few pieces of text containing the new word. Their technique involved finding "statistical synonyms" for the new word, and basing the probabilities for the new word on the statistical synonyms.

However, if a *class* n-gram model is employed, the task of adding a new word is simplified whenever the word belongs to a pre-existing word class. In a class n-gram model, words are collapsed into classes (which are generally syntactically and/or semantically motivated). If we want to add a new word to a class n-gram, and the word belongs to a class that is modeled in the language model, then our task is significantly easier than were we adding the word to a word n-gram. In the SUMMIT system, which uses class n-gram language models, adding a new word to an existing class is straightforward:

- the word is added to the appropriate class and
- the unigram class-conditional probabilities (i.e., $P(word \mid class)$) for the appropriate class are updated.

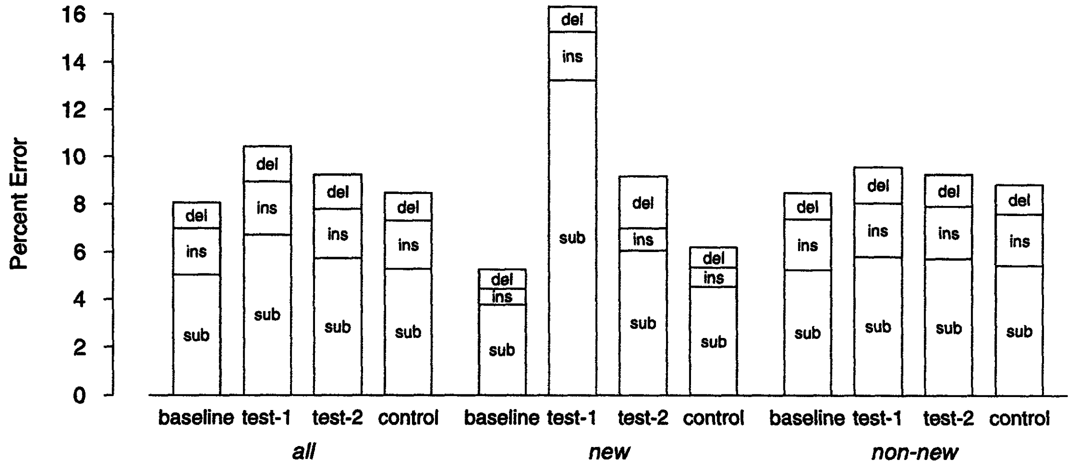
Adding a word to a class is trivial if we know its class; we simply add it to the list of words for the class. Updating the class-conditional unigram word probabilities can be more difficult. The optimal way to set these probabilities is by observing a (large) set of training text and counting the number of times that a class is represented by the particular new word.

However, in the absence of training text containing the new word, we can still update the class-conditional unigram probabilities in an ad hoc fashion. For example, to add a new city name, we could set the new city name's class-conditional unigram probability to be the minimum or average probability for words in the class. The advantage of the *class* n-gram language model for incorporating new words is that only *one* probability needs to be estimated for each new word: its class-conditional unigram probability. The myriad of n-gram probabilities (that depend on a large number of sequences of words) do not need to be updated because the class mechanism shares the probabilities for *contexts.* Of course, if a new word does not belong to an existing word class, the problem of adding it to the language model is similar to the problem of adding it to a word n-gram. Because the word cannot be added to an existing class, it cannot share the contextual probabilities of an existing class.

We performed a series of experiments to evaluate how well new words could be added to SUMMIT's class n-gram language models. We used four systems: a baseline system, the test-1 system, the test-2 system, and a control system. The systems differed only in the particular class 4-gram language model used. The baseline system was the system fully trained on the new words. The test-1 and test-2 systems were both trained on the reduced-vocabulary training set. The test-1 system set the class-conditional unigram probability for the new words to the *minimum* value for the appropriate classes. In contrast, the test-2 system set the class-conditional unigram probability to the *average* value. Finally, the control system was trained on the new words, but the training set size was comparable to the size of the reduced-vocabulary training set used for the test-1 and test-2 systems. The control system served as a control for training set size.

New words that did not belong to the pre-existing word classes were modeled crudely within the language model. The δ-smoothing mechanism (Section 3.6), which smoothes the interpolated class n-gram probabilities with uniform unigram word probabilities, was used to assign probabilities for the words that did not fit into classes. With the value $\delta = 0.02$ used, the probability of all words was at least $\delta/||\mathcal{V}||$, where $||\mathcal{V}|| = 2,461$ was the size of the vocabulary. New words that belonged to classes were modeled better than those that did not, but probabilities could be assigned to all new words.

Figure 5-21 summarizes the performances of the baseline, test-1, test-2, and control systems evaluated on the full test set $\mathcal{S}$.Compared to the baseline system, the test-1 and test-2 systems had worse overall word-error rates. The error rate for test-1 was 2.4% (factor of 1.3) worse, and that of test-2 was 1.2% (factor of 1.1) worse. On the set of new words, the test-1 and test-2 systems were significantly worse than the baseline system, with error rates 11.1% (factor of 3.1) and 3.9% (factor of 1.7) higher than the baseline system, respectively. This indicates that the new words are not modeled in the language model nearly as well as the non-new words. The test-2 system, with its higher class-conditional unigram probabilities for the new words, did perform much better than the test-1 system on the new words. We felt that part of the explanation for the relatively poor performance of the test-1 and test-2 systems was due to the reduced training set size. However, examination of the performance of the control system revealed that only a small part of the degradation was likely due to the reduction in training set size. The



(a)

	n_{total}	*sub*		*ins*		*del*		*total*	
		n	%	n	%	n	%	n	%
all									
baseline	12,707	640	5.0	247	1.9	136	1.1	1,023	8.1
test-1		855	6.7	282	2.2	188	1.5	1,325	10.4
Δ		+215	+1.7	+35	+0.3	+52	+0.4	+302	+2.4
test-2		729	5.7	259	2.0	184	1.4	1,172	9.2
Δ		+89	+0.7	+12	+0.1	+48	+0.4	+149	+1.2
control		671	5.3	256	2.0	149	1.2	1,076	8.5
Δ		+31	+0.2	+9	+0.1	+13	+0.1	+53	+0.4
new									
baseline	1,618	61	3.8	11	0.7	13	0.8	85	5.3
test-1		214	13.2	33	2.0	17	1.1	264	16.3
Δ		+153	+9.5	+22	+1.4	+4	+0.2	+179	+11.1
test-2		98	6.1	15	0.9	35	2.2	148	9.1
Δ		+37	+2.3	+4	+0.2	+22	+1.4	+63	+3.9
control		73	4.5	13	0.8	14	0.9	100	6.2
Δ		+12	+0.7	+2	+0.1	+1	+0.1	+15	+0.9
non-new									
baseline	11,089	579	5.2	236	2.1	123	1.1	938	8.5
test-1		641	5.8	249	2.2	171	1.5	1,061	9.6
Δ		+62	+0.6	+13	+0.1	+48	+0.4	+123	+1.1
test-2		631	5.7	244	2.2	149	1.3	1,024	9.2
Δ		+52	+0.5	+8	+0.1	+26	+0.2	+86	+0.8
control		598	5.4	243	2.2	135	1.2	976	8.8
Δ		+19	+0.2	+7	+0.1	+12	+0.1	+38	+0.3

(b)

Figure 5-21: Updating the language model. The test-1 language model had class-conditional word unigram probabilities $P(w \mid c(w))$ set to the *minimum* value for the class $c(w)$, whereas for test-2 they were set to the *average* value.

control system performed only slightly worse than the baseline system despite having a 17% smaller training set.

Even though the nature of the class 4-gram language model made adding most of the words straightforward, the shortfall in performance of the test-1 and test-2 systems is evidence of the importance of a well-trained language model. However, we were able to incorporate new words into our class 4-gram model without *any* training text containing them. Because the class nature of a class n-gram language model shares much of the contextual information, we were able to add new words that fit into the predefined classes simply by adding them to the class list and setting a single class-conditional unigram probability. We saw that how this probability is set can have a large impact on performance as evidenced by the performance difference of the test-1 and test-2 systems. However, the problem of finding a suitable class-conditional unigram probability for a classified word is considerably easier than estimating contextual probabilities for words in a non-class n-gram language model. Words that did not fit into classes were assigned the context-independent probability $\delta/||\mathcal{V}||$. However, for the set of simulated new words used in this study, 76% of them fit into the predefined word classes. Weighted by word frequency, fully 97% fit into classes. Thus, we did no have to rely on the crude context-independent uniform unigram probability very often. Therefore, we find that the *class* n-gram language model enables new-word incorporation with reasonable performance when new words fit into predefined word classes.

5.5.4 Summary of Learning Issues

In this section we examined how well our system could incorporate new words without *any* training data containing them. Specifically, we examined the vocabulary independence of context-independent acoustic models, how well untrained pronunciation models for new words performed, and how well we could update the class 4-gram language model. We found that each of these three sets of models could be used directly or updated in order to recognize new words with reasonable levels of success without retraining on *any* spoken utterances of the new words.

Table 5-13 shows the word-error rates for the different system configurations. Word-

acoustic	lexical	language	S	S	S'	S'
models	models	models	all	new	all	new
B	B	B	8.1	5.3	9.1	7.5
R	R	R	26.7	125.0	19.8	132.8
R	B	B	*8.9	*9.2	9.1	8.6
B	R'	B	8.1	5.6	9.1	7.5
B	B	R'	9.2	9.2	9.9	8.8
R	R'	R'	*10.8	*16.3	10.0	12.0

Table 5-13: Summary of learning new words. Word-error rates (%) were evaluated for all words and new words only over two test sets, S and S'. S is the full test set, and S' is the reduced test set that does not require the poorly trained [o^y] and male [ŋ̩] acoustic models. Error rates marked with (*) were affected by the poorly trained acoustic models. In the table, B indicates baseline (fully trained) models, R indicates reduced-vocabulary models, and R' indicates updated reduced-vocabulary models. Thus, BBB is the full baseline system, RRR is the original reduced-vocabulary system, and $RR'R'$ is the reduced-vocabulary system with new words incorporated.

error rates are given over all words and over just new words, for two different test sets, S and S'. S is the full test set, and S' is the reduced test set that does not require the poorly trained [o^y] and male [ŋ̩] models.

In the table, B indicates baseline models, R indicates reduced-vocabulary models, and R' indicates updated reduced-vocabulary models. The top row (BBB) is the baseline system that is fully trained on the set of new words. The second row (RRR) is the original, reduced-vocabulary system that experiences new words. The next three rows (RBB, $BR'B$, and BBR') are the systems one original or updated set of models. Finally, the last row ($RR'R'$) is the complete reduced-vocabulary system with new words incorporated.

The lexical models were least in need of training data. The lexical models could be constructed with base-form pronunciations, such as available in a dictionary, and the application of phonological rules. The trainable component of our lexical models, the lexical arc weights, did not benefit from training, and thus do not appear important for learning new words.

The context-independent acoustic models could have benefited from some additional training data, but that might not be the case if they were originally trained on enough data. We found that when trained on our reduced-vocabulary training set some of the acoustic models were inadequately trained because they were too rare. Certainly this is unacceptable if we wish to have a system that is capable of incorporating new words. If not all acoustic models that could be used by new word lexical models have enough task-specific training data, those data should be supplemented with task-independent training data (e.g., TIMIT). With context-*independent* acoustic models this should be feasible. With context-*dependent* acoustic models a very large number of models, well above that required for the original vocabulary, would need to be trained for possible inclusion in new words. Thus, context-independent models are more conducive to the incorporation of new words.

With the use of a *class* n-gram language model we were able to incorporate new words without any additional training text, although such text would have boosted performance. We found that the nature of our class n-gram model was such that we could add a new word to a class by specifying a single probability, the class-conditional unigram probability, and allow the contextual sharing of the language model to capture the contextual probabilities. Words that did not fit into the predefined classes were modeled with a crude unigram probability. We feel that the use of a *class* language model was important to achieving reasonable recognition performance with no additional training data.

When we combine the new acoustic, lexical, and language models, we have a system that was trained in the absence of new words. This system achieved a 12.0% word-error rate compared to the baseline system's 7.5% (on the reduced evaluation set). That is a factor of 1.6 times worse than the baseline system that was trained extensively on the new words. The 12.0% error rate shows what is possible using very simple new-word incorporation technique and no additional data. Furthermore, Table 5-13 shows that we can greatly reduce the error rate of the original reduced-vocabulary system (RRR) by incorporating new words ($RR'R'$ system), dropping the word-error rate by nearly 10%. With more sophisticated techniques, possibly combined with a small amount of data containing new words, the performance could only improve. The results of this section show that new words can be readily learned to a degree by a system without any training data in a supervised manner.

5.6 Summary

In Chapter 2 we demonstrated that new, out-of-vocabulary words occur in a wide-variety of tasks no matter how large a system vocabulary is used. The new-word rate is a function of the characteristics of the task as well as the vocabulary size. However, although this study showed us that new words occur, it did not tell us anything about their effect on an actual continuous-speech recognition system. In this chapter we have *quantified* the effects of new words using carefully controlled experiments. Through the use of a baseline system and a reduced-vocabulary system we have been able to examine the effects of (simulated) new words in terms of performance degradation and increase in computational complexity. Additionally, in looking forward to a system that is capable of new-word detection, we have examined some of the issues related to *learning*, or incorporating, new words so that they become part of a system's working vocabulary.

In examining the effect new words have on *recognition performance,* we found that the SUMMIT system experiences about 1.5 word errors per new word. This corresponds to about 1.2 errors associated with the new word itself, which implies that the recognizer often substitutes more than one in-vocabulary word per new word. More importantly, about 0.3 *in-vocabulary* words are misrecognized per new word. In other words, the occurrence of a single new word can cause a ripple effect to nearby in-vocabulary words. In our analysis, we found that in-vocabulary words adjacent to new words were most affected. We examined not only single, isolated new words but also multiple new words per utterance including sequences of new words and disjoint new words. We found that sequences of new words did not present any additional problems compared to isolated new words; in effect they were relatively long single new words. However, we found that with disjoint new words, in-vocabulary words between them were misrecognized at a much higher rate than other words, although this result may not generalize to other situations due to the relatively fixed structure of many ATIS utterances.

In examining the effect new words have on *computational complexity* of the recognition search, we learned that a single new word per utterance increased the amount of computation per second of speech by nearly a factor of four. We developed graphical displays based on the word graphs presented in Chapter 4 that show the increase in recognizer "uncertainty" in the vicinity of new words as it attempts to account for new words with many combinations of in-vocabulary words. We developed a measure of this uncertainty, called the active-word count, that appears to be strongly correlated with the *location* of new words, and thus may be a useful tool in performing new-word recognition. Finally, and most importantly, we showed that our word graph computation and representation are insensitive to the *position* of new words within utterances. With a search algorithm that opens the search space into a tree (e.g., A* N-best algorithm) we would expect the position of a new word to affect the computational complexity. However, because our A* word graph algorithm does not open the search space into a tree during lexical access, computation is not dependent on the position of a new word within an utterance.

In examining the issues related to *learning* new words, we showed that it was possible, with straightforward techniques, to add a large number of new words to a system without additional training material. Specifically, we separately updated context-independent acoustic models, lexical (pronunciation) models, and class n-gram models. For each of these models, we measured the shortfall in performance compared to a baseline system that was fully trained on the new words. We found that, with our system, the lexical models benefited the least from training data containing new words, followed by the acoustic and language models. We discovered that, even with vocabularies on the order of 1,500 words with training sets containing nearly 20,000 words, it is possible for some of the more rare context-*independent* models to suffer from sparse data and be inadequately trained. In building a system capable of new-word learning, this problem would certainly have to be rectified, probably with the use of supplementary vocabulary-independent training data to guarantee that *all* acoustic models are adequately trained. With regard to updating language models, we found that a *class* n-gram language model could be updated without any additional training data because classes share the contextual modeling information. Finally, we were able to construct a complete system capable of recognizing new word it had never been trained on, albeit with a word-error rate on the new words that was 1.6 times as large as the word-error rate of a *fully* trained system.

Overall, this chapter represents a quantitative study of the new-word problem within the context of a continuous-speech recognition system. Combined with Chapter 2 in which we demonstrated the *frequency* of new words, we now understand the *effects* of new words on a recognizer in terms of performance degradation and increase in computational complexity. Together, these two chapters demonstrate that the new-word problem is a serious problem that needs significantly more attention in order to solve it through the use of new-word detection and new-word learning. We leave the problem of new-word detection for future work, but have looked briefly at the issues of learning new words. We showed that with straightforward techniques, new words can be learned without requiring training data containing them, given that we are supplied with a spelling, a phonemic base-form, and a word class.

Chapter 6

Conclusion

This thesis represents an introduction and characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding. We feel that the new-word problem is one that cannot be ignored if we are to produce spoken language systems that can function successfully in the real world. New words can occur for several reasons, including mismatch between system training and actual use, the imperfect nature of vocabulary determination, the invention of new words, and the fact that users do not know the exact limits of a system's vocabulary.

Much of the research in speech recognition and understanding has ignored the new-word problem. This is partly because common recognition and understanding tasks have limited the frequency of new words, either explicitly or implicitly, to the point that they have only minimal effect on system performance. By examining the common tasks used within the ARPA speech recognition and understanding community, the artificial reduction or complete elimination of the new-word problem is apparent. The Resource Management task used a completely closed vocabulary for the scripts used to collect the read speech. More recently, the current large-vocabulary Wall Street Journal task has been evaluated in a way that artificially reduces the frequency of new words. Specifically, the evaluation utterances were selected so as to contain at most the 64,000 most frequent words within the corpus. Even though systems were tested with smaller vocabularies (20,000 and 40,000 words), this vocabulary filtering significantly affected the frequency of new words. Even within the Air Travel Information Service

(ATIS) task we may see fewer new words than we would expect given real users solving travel-planning problems. The way the ATIS data has been collected, with users solving predetermined travel problems or scenarios, may implicitly limit users' vocabularies. In general, we expect that new words are a bigger problem than such specific tasks might indicate, and that new words could be a serious impediment to the eventual deployment of useful spoken language systems without significant improvement in the area of new-word detection and learning.

6.1 Summary

In the first part of this thesis (Chapter 2) we measured the frequency of new words in a corpus-based study of new words. By examining several corpora from widely varying tasks (including spontaneously spoken human/computer interaction, actual directory assistance telephone calls, human/human telephone conversation, and newspaper texts), for three different languages, we were able to characterize tasks in terms of vocabulary size and new-word rate (frequency). We found that the vocabulary size and new-word rate characteristics could be used to cluster the corpora, using factors such as domain restrictiveness, speech versus writing, and communication with a human versus with a computer. The clustering results appeared to be language independent, at least for the languages we examined (English, French, and Italian).

We found that some tasks can require very large vocabularies—on the order of 100,000 words—to reduce the new-word rate to 1%. In contrast, some human/computer interactive problem-solving tasks can achieve the same new-word rate with much smaller vocabularies. However, the question remains as to the realism of the data collection methods when prescribed scenarios are used. The use of scenarios may have a large effect on the words people use when interacting with the (data collection) systems because the scenarios restrict the problem-solving domain. While a new-word rate of 1% may seem low, especially if we are concerned only with *recognition* accuracy such as measured by word-error rate, it is likely too high for *understanding.* We found that a 1% new-word rate can correspond to 17% of utterances containing at least one new word.

If understanding is the goal, having nearly one utterance in five containing a new word likely to cause understanding errors could be a real problem. In general, by examining vocabulary sizes and new-word rates for a diverse set of corpora, we concluded that new words are frequent enough that they cannot be ignored.

The new-word study of Chapter 2 examined only the orthographic transcriptions (i.e., text) of the corpora, and thus, was completely independent of any specific recognition or understanding system. In order to gauge the impact new words have on system behavior, we examined the new-word problem within the context of a continuous-speech recognition system. However, in performing this study it became apparent that information contained in recognizer N-best lists was insufficient for our purposes.

In Chapter 4, we introduced an algorithm for computing word graphs based on the A* N-best algorithm. We initially developed word graphs so that we could gain a glimpse into the individual words, their acoustic scores, and their time alignments during the recognition search. Using word graphs, we introduced two exploratory data analysis tools, word lattice displays and the active word count, which we utilized in our recognizer-based study of the new-word problem.

We found that word graphs were useful for continuous-speech recognition irrespective of the new-word problem. In general, word graphs are an alternative to the more traditional N-best lists. Word graphs contain N-best recognizer hypotheses, and all possible alignments thereof, down to a predetermined relative score threshold in a relatively compact representation that can be computed efficiently.

Finally, in Chapter 5, we examined in detail the interaction of new words with the SUMMIT continuous-speech recognition system within the ATIS domain. Through the use of a baseline system and a reduced-vocabulary system we were able to examine the effect of new words in terms of both word accuracy and computational complexity. By comparing the output of the two systems on the same sets of utterances we were able to separate the effects due to new words from the normal system behavior when there were no new words.

In terms of word accuracy, we found that each new word was responsible for about 1.5 errors on average. Of these errors, about 0.3 errors were to *in-vocabulary* words that would otherwise have been correctly recognized had there not been a nearby new word. The other 1.2 errors per new word were due to the in-vocabulary words the recognizer substituted in place of a new word. Examining these substitutions for new words, we found that *within-class* substitutions will likely be a significant problem for speech *understanding.* If a recognizer substitutes in-vocabulary words that make no sense, the natural language understanding component will likely have difficulty understanding the recognizer's output, hopefully alerting the user to the out-of-vocabulary word. On the other hand, if there are no other recognition errors for an utterance, a within-class substitution will be interpreted by the natural language component, albeit incorrectly. Depending on the level of system feedback, such a misunderstanding may cause subsequent user and system confusion in an interactive dialog. For the ATIS task and the particular simulated new words in our study, we found that for 45% of new words our recognizer produced such a within-class substitution.

Using tools developed in this thesis and based on our word graph representation, we were able to quantify the effect of new words on the computational complexity of the recognition search. We found that new words had a significant effect on the amount of computation per second of speech as measured by the number of word edges in a word graph. By examining *word lattice displays,* it was evident that this increase was due to the relatively large number of in-vocabulary words that a recognizer proposes to account for the acoustics of a new word. Further, we introduced the *active word count* as a local measure of computational complexity. We found that high values of the count were correlated with the *location* of new words. We hypothesize that the active word count could be a useful measure for the new-word detection problem.

Finally, we examined the problem of incorporating new words into a recognition system. Specifically, we added new words to our system in a supervised manner without retraining any system components using utterances or text containing the new word. We then measured the shortfall in recognition performance compared to the baseline condition in which all the new words were part of the system's full training.

We found that context-independent phonetic models were helpful in modeling unforeseen new words provided that all phonetic models were adequately trained. We found that starting with a phonemic base-form pronunciation of a new word we were able to incorporate pronunciation models for new words into our system even though it uses relatively complex pronunciation networks with weighted alternative pronunciations. We found that for new words it was not important to train the pronunciation weights, meaning that we could create pronunciations of new words in a supervised manner without any new word utterances. Finally, we found that we could add new words to our word-class n-gram language models by explicitly adding words (by hand) to the system's predefined word classes. For our ATIS system, this was a relatively straightforward process, resulting in an updated language model that could be used to recognize new words. However, words that did not fit neatly into the system's predetermined word classes were only crudely modeled within the language model. Such unclassified words would likely benefit from some examples of the new words in context. Nevertheless, using straightforward techniques we were able to add new words to our continuous-speech recognition system with only a moderate increase in error rate without retraining on *any* additional training utterances containing them.

In summary, the new-word problem is a very wide-ranging problem that touches upon virtually all aspects of speech recognition and understanding. In this thesis we have studied the magnitude and scope of the problem in terms of the frequency of new words and their effects on an actual speech recognition system. We feel that the new-word problem has not received the attention it deserves, and this thesis is an attempt to motivate others to work on it.

6.2 Future Work

The size of the new-word problem is such that it could not be explored and solved within the scope of this thesis. There is significant opportunity to extend the work of this thesis and previous new-word research. To our knowledge, the impact of new words on speech *understanding* has not been adequately addressed. We are interested in carrying out controlled understanding experiments similar to the recognition experiments of this thesis.

While there has been some recent research on the new-word detection problem, we believe that the problem is far from being solved. Most previous detection techniques involve modeling the acoustics of new words using (loosely constrained) sequences of phonetic models. Perhaps additional detection cues can be incorporated that would improve detection performance. In particular, local computational complexity measures such as our active word count may provide useful information that could be combined with the more traditional acoustic modeling approach. Furthermore, it is possible that the use of natural language constraints could aid in the new-word detection problem.

We have not examined the problem of adding new words to a system when there are utterances containing new words available. Such utterances could be used to refine the acoustic/pronunciation modeling of new words as evidenced by the work of Asadi et al. [3–5]. Examples of new words in context could be used to refine the language modeling of new words within n-gram models, particularly when the new words do not fit neatly into the system's predefined word classes. A technique for identifying similar words, or "statistical synonyms," such as that of Jelinek et al. [34] could help in such cases. Finally, we did not address the problem of adding new words to a natural language understanding system. Presumably, it might be possible for such a system to automatically determine some semantics of new words. For example, in the utterance "I want to fly to *new-word* on the 16th" the understanding component could reasonably hypothesize that the new word was a destination for air travel. Adding new words to the understanding component of spoken language systems remains an important problem.

Finally, the observation that a potentially large fraction of new words are derivations (e.g., inflections or concatenations) of in-vocabulary words is an important one. If we were to change the unit of speech recognition and understanding from the word level to the morpheme level, it might be possible to reduce the number of new words. A morpheme-based system may be able to recognize and understand novel words (e.g., "ungood") by combining word roots, prefixes, and suffixes.

Bibliography

[1] F. Alleva and K.-F. Lee, "Automatic new word acquisition: Spelling from acoustics," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 266–270, Harwichport, MA, October 1989.

[2] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic detection of new words in a large-vocabulary continuous speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 125–128, Albuquerque, NM, April 1990.

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

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

[5] A. O. Asadi and H. C. Leung, "New-word addition and adaptation in a stochastic explicit-segment speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 5, pp. 642–645, Minneapolis, April 1993.

[6] Association for Computational Linguistics Data Collection Initiative, "CD-ROM I," September 1991.

[7] X. Aubert, C. Dugast, H. Ney, and V. Steinbiss, "Large vocabulary continuous speech recognition of the Wall Street Journal data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 129–132, Adelaide, April 1994.

[8] A. Barr, E. Feigenbaum, and P. Cohen, *The Handbook of Artificial Intelligence.* William Kaufman, Los Altos, CA, 1981.

[9] R. A. Becker, J. M. Chambers, and A. R. Wilks, *The New S Language: a Programming Environment for Data Analysis and Graphics.* Wadsworth and Brooks/Cole, Pacific Grove, CA, 1988.

[10] H. Bonneau-Maynard, J.-L. Gauvain, D. Goodine, L. F. Lamel, J. Polifroni, and S. Seneff, "A French version of the MIT-ATIS system: Portability issues," in *Proc. European Conf. Speech Communication and Technology*, pp. 2059–2062, Berlin, September 1993.

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

[12] B. Chigier and J. Spitz, "Are laboratory databases appropriate for training and testing telephone speech recognizers?," in *Proc. Int. Conf. Spoken Language Processing*, pp. 1017–1020, Kobe, November 1990.

[13] Y. Chow and R. Schwartz, "The N-best algorithm," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 199–202, Harwichport, MA, October 1989.

[14] Y. L. Chow, M. O. Dunham, 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. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 89–92, Dallas, April 1987.

[15] Y. L. Chow, M. O. Dunham, 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 A. Waibel and K.-F. Lee (eds.), *Readings in Speech Recognition*, pp. 596–599. Morgan Kaufmann, San Mateo, CA, 1990.

[16] K. W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," in *Proc. 2nd Conf. on Applied Nat. Lang. Processing*, pp. 136–143, Austin, TX, February 1988.

[17] D. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and L. Shriberg, "Expanding the scope of the ATIS task: The ATIS-3 corpus," in *Proc. ARPA Human Lang. Tech. Workshop*, Princeton, March 1994.

[18] V. Digalakis, *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*, Ph.D. thesis, Boston University, Boston, June 1992.

[19] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "A dynamical system approach to continuous speech recognition," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 253–257, Pacific Grove, CA, February 1991.

[20] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "A dynamical system approach to continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 289–292, Toronto, May 1991.

[21] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans. Speech and Audio Processing*, 1(4):431–442, October 1993.

[22] G. Flammia, J. Glass, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, "Porting the bilingual VOYAGER system to Italian," in *Proc. Int. Conf. Spoken Language Processing*, pp. 911–914, Yokohama, September 1994.

[23] J.-L. Gauvain, L. F. Lamel, G. Adda, and M. Adda-Decker, "The LIMSI Nov93 WSJ system," in *Proc. ARPA Spoken Lang. Sys. Tech. Workshop*, Princeton, March 1994.

[24] J.-L. Gauvain, L. F. Lamel, and M. Eskénazi, "Design considerations and text selection for BREF, a large French read-speech corpus," in *Proc. Int. Conf. Spoken Language Processing*, pp. 1097–1100, Kobe, November 1990.

[25] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 517–520, San Francisco, March 1992.

[26] D. Goodine, S. Seneff, L. Hirschman, and M. Phillips, "Full integration of speech and language understanding in the MIT spoken language system," in *Proc. European Conf. Speech Communication and Technology*, pp. 845–848, Genoa, September 1991.

[27] I. L. Hetherington, M. S. Phillips, J. R. Glass, and V. W. Zue, "A* word network search for continuous speech recognition," in *Proc. European Conf. Speech Communication and Technology*, pp. 1533–1536, Berlin, September 1993.

[28] I. L. Hetherington and V. W. Zue, "New words: Implications for continuous speech recognition," in *Proc. European Conf. Speech Communication and Technology*, pp. 2121–2124, Berlin, September 1993.

[29] L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, K. Hunicke-Smith, D. Pallett, C. Pao, P. Price, and A. Rudnicky, "Multi-site data collection for a spoken language corpus," in *Proc. Int. Conf. Spoken Language Processing*, pp. 903–906, Banff, October 1992.

[30] S. Hunnicutt, H. Meng, S. Seneff, and V. Zue, "Reversible letter-to-sound sound-to-letter generation based on parsing word morphology," in *Proc. European Conf. Speech Communication and Technology*, pp. 763–766, Berlin, September 1993.

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

[32] F. Jelinek, "Fast sequential decoding using a stack," *IBM J. Res. Develop.*, 13(6):675–685, November 1969.

[33] F. Jelinek, "Self-organized language modeling for speech recognition," in A. Waibel and K.-F. Lee (eds.), *Readings in Speech Recognition*, pp. 450–506. Morgan Kaufmann, San Mateo, CA, 1990.

[34] F. Jelinek, R. Mercer, and S. Roukous, "Classifying words for improved statistical language models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 621–624, Albuquerque, NM, April 1990.

[35] P. Kenny, R. Hollan, V. Gupta, M. Lennig, P. Mermelstein, and D. O'Shaughnessy, "A*-admissible heuristics for rapid lexical access," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 689–692, Toronto, May 1991.

[36] P. Kenny, P. Labute, Z. Li, and D. O'Shaughnessy, "New graph search techniques for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 553–556, Adelaide, April 1994.

[37] V. Khazatsky, *Speech Recognition: Personalization of Vocabulary (Information Theoretical Approach)*, Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1985.

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

[39] L. F. Lamel, J.-L. Gauvain, and M. Eskénazi, "BREF, a large vocabulary spoken corpus for French," in *Proc. European Conf. Speech Communication and Technology*, pp. 505–508, Genoa, September 1991.

[40] H. C. Leung, I. L. Hetherington, and V. W. Zue, "Speech recognition using stochastic explicit-segment modeling," in *Proc. European Conf. Speech Communication and Technology*, pp. 931–934, Genoa, September 1991.

[41] H. C. Leung, I. L. Hetherington, and V. W. Zue, "Speech recognition using stochastic segment neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 613–616, San Francisco, March 1992.

[42] H. M. Meng, S. Seneff, and V. W. Zue, "Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation," in *Proc. ARPA Human Lang. Tech. Workshop*, Princeton, March 1994.

[43] H. M. Meng, S. Seneff, and V. W. Zue, "Phonological parsing for reversible letter-to-sound/sound-to-letter generation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 1–4, Adelaide, April 1994.

[44] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-vocabulary dictation using SRI's DECIPHER speech recognition system: Progressive search techniques," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 319–322, Minneapolis, April 1993.

[45] H. Murveit and R. Moore, "Integrating natural language constraints into HMM-based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 573–576, Albuquerque, NM, April 1990.

[46] L. Nguyen, R. Schwartz, F. Kubala, G. Chou, C. Lapre, Y. Zhao, J. Makhoul, G. Zavaliagkos, and A. Anastasakos, "Spoke 9: Spontaneous WSJ dictation," in *Proc. ARPA Spoken Lang. Sys. Tech. Workshop*, Princeton, March 1994.

[47] N. J. Nilsson, *Principles of Artificial Intelligence.* Morgan Kaufmann, San Mateo, CA, 1980.

[48] M. Oerder and H. Ney, "Word graphs: An efficient interface between continuous-speech recognition and language understanding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 119–122, Minneapolis, April 1993.

[49] D. O'Shaughnessy, *Speech Communication: Human and Machine.* Addison-Wesley, Reading, MA, 1987.

[50] M. Ostendorf, I. Bechwati, and O. Kimball, "Context modeling with the stochastic segment model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 389–392, San Francisco, March 1992.

[51] M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz, and J. R. Rohlicek, "Integration of diverse recognition methodologies through reevaluation of N-best sentence hypotheses," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 83–87, Pacific Grove, CA, February 1991.

[52] M. Ostendorf and S. Roukos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):1857–1869, December 1989.

[53] D. S. Pallett, J. G. Fiscus, W. M. Fisher, and J. S. Garofolo, "Benchmark tests for the DARPA spoken language program," in *Proc. ARPA Human Lang. Tech. Workshop*, Plainsboro, NJ, March 1993.

[54] D. S. Pallett, W. M. Fisher, and J. G. Fiscus, "Tools for the analysis of benchmark speech recognition tests," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 97–100, Albuquerque, April 1990.

[55] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 357–362, Harriman, NY, February 1992.

[56] M. Phillips, J. Glass, and V. Zue, "Modelling context dependency in acoustic-phonetic and lexical representations," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 71–76, Pacific Grove, CA, February 1991.

[57] M. Phillips and V. Zue, "Automatic discover of acoustic measurements for phonetic classification," in *Proc. Int. Conf. Spoken Language Processing*, pp. 795–798, Banff, October 1992.

[58] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-word Resource Managment database for continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 651–654, New York, April 1988.

[59] P. J. Price, "Evaluation of spoken language systems: the ATIS domain," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 91–95, Hidden Valley, CA, June 1990.

[60] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[61] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, 77(2):257–286, February 1989.

[62] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 701–704, Toronto, May 1991.

[63] R. Schwartz, S. Austin, F. Kubala, J. Makhoul, L. Nguyen, P. Placeway, and G. Zavaliagkos, "New uses for the N-best sentence hypotheses within the BYBLOS speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 1–4, San Francisco, March 1992.

[64] R. Schwartz and Y.-L. Chow, "The N-best algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 81–84, Albuquerque, NM, April 1990.

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

[66] S. Seneff, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, H. Leung, M. Phillips, J. Polifroni, and V. Zue, "Development and preliminary evaluation of the MIT ATIS system," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 88–93, Pacific Grove, CA, February 1991.

[67] F. K. Soong and E.-F. Huang, "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 705–708, Toronto, May 1991.

[68] J. Spitz, "Collection and analysis of data from real users: Implications for speech recognition/understanding systems," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 164–169, Pacific Grove, CA, February 1991.

[69] B. Suhm, M. Woszczyna, and A. Waibel, "Detection and transcription of new words," in *Proc. European Conf. Speech Communication and Technology*, pp. 2179–2182, Berlin, September 1993.

[70] A. Viterbi, "Error bounds for convolutional codes and an asymptotic optimal decoding algorithm," *IEEE Trans. Inform. Theory*, IT-13:260–269, April 1967.

[71] G. Zavaliagkos, T. Anastasakos, G. Chou, F. Kubala, C. Lapre, J. Makhoul, L. Nguyen, R. Scwhartz, and Y. Zhao, "BBN hub system and results," in *Proc. ARPA Spoken Lang. Sys. Tech. Workshop*, Princeton, March 1994.

[72] V. Zue, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, M. Phillips, J. Polifroni, and S. Seneff, "The MIT ATIS system: February 1992 progress report," in *Proc. DARPA Speech and Nat. Lang. Workshop*, pp. 84–88, Harriman, NY, February 1992.

[73] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Integration of speech recognition and natural language processing in the MIT VOYAGER system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 713–716, Toronto, May 1991.

[74] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, and S. Seneff, "The VOYAGER speech understanding system: Preliminary development and evaluation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 73–76, Albuquerque, NM, April 1990.

[75] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff, "The SUMMIT speech recognition system: Phonological modelling and lexical acess," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 49–52, Albuquerque, NM, April 1990.

[76] V. Zue, J. Glass, M. Phillips, and S. Seneff, "Acoustic segmentation and phonetic classification in the SUMMIT system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 389–392, Glasgow, May 1989.

[77] V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goddeau, J. Glass, and E. Brill, "The MIT ATIS system: December 1993 progress report," in *Proc. ARPA Spoken Lang. Sys. Tech. Workshop*, Princeton, March 1994.

[78] V. W. Zue, "Toward systems that understand spoken language," *IEEE Expert*, pp. 51–59, February 1994.