DECLARATION OF GORDON MACPHERSON

I, Gordon MacPherson, am over twenty-one (21) years of age. I have never been convicted of a felony, and I am fully competent to make this declaration. I declare the following to be true to the best of my knowledge, information, and belief:

1. I am the Director, Board Governance & Policy Development at The Institute of Electrical and Electronics Engineers, Incorporated ("IEEE").
2. IEEE is a neutral third party in this dispute.
3. I am not being compensated for this declaration and IEEE is only being reimbursed for the cost of the article I am certifying.
4. Among my responsibilities as Director of Board Governance & Policy Development, I act as a custodian of certain records for IEEE.
5. I make this declaration based on my personal knowledge and information contained in the business records of IEEE.
6. As part of its ordinary course of business, IEEE publishes and makes available technical articles and standards. These publications are made available for public download through the IEEE digital library, IEEE Xplore.
7. It is the regular practice of IEEE to publish articles and other writings including article abstracts and make them available to the public through IEEE Xplore. IEEE maintains copies of publications in the ordinary course of its regularly conducted activities.
8. The articles below have been attached as Exhibit A-Exhibit E to this declaration:

Exhibit A	I. Bazzi and J. Glass, "Heterogeneous Lexical Units for Automatic Speech Recognition: Preliminary Investigations," 2000 IEEE International Conference on Acoustics, Speech, And Signal Processing. Proceedings (Cat. No.00CH37100), Istanbul, Turkey, 2000, pp. 1257-1260 vol.3, doi: 10.1109/ICASSP.2000.861804.
Exhibit B	J. Glass, J. Chang and M. McCandless, "A Probabilistic Framework for Feature-Based Speech Recognition," Proceeding of Fourth International Conference on Spoken Language Processing ICSLP '96, Philadelphia, PA, USA, 1996, pp. 2277-2280 vol.4, doi: 10.1109/ICSLP.1996.607261
Exhibit C	J. R. Glass, T. J. Hazen and I. L. Hetherington, "Real-Time Telephone-Based Speech Recognition in the Jupiter Domain," 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing

	Proceedings ICASSP99 (Cat. No. 99CH36258), Phoenix, AZ, USA, 1999, pp. 61-64 vol.1, doi: 10.1109/ICASSP.1999.758062.
Exhibit D	Jilei Tian, J. Nurminen and I. Kiss, "Optimal Subset Selection from Text Databases," Proceedings (ICASSP '05) IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, Philadelphia, PA, USA, 2005, pp. I/305-I/308 Vol. 1, doi: 10.1109/ICASSP.2005.1415111.
Exhibit E	Su-Lin Wu, M. L. Shire, S. Greenberg and N. Morgan, "Integrating Syllable Boundary Information into Speech Recognition," 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 1997, pp. 987-990 vol.2, doi: 10.1109/ICASSP.1997.596105.

9. I obtained a copy of Exhibit A- Exhibit E through IEEE Xplore, where it is maintained in the ordinary course of IEEE's business. Exhibit A-Exhibit E are true and correct copies of the Exhibits as they existed on or about June 12, 2025.
10. The article and abstract from IEEE Xplore show the date of publication. IEEE Xplore populates this information using the metadata associated with the publication.
11. I. Bazzi and J. Glass, "Heterogeneous Lexical Units for Automatic Speech Recognition: Preliminary Investigations," was published in 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (Cat. No.00CH37100), held on June 5-9, 2000, in Istanbul, Turkey. Copies of the conference proceedings were made available no later than the last day of the conference. This article was added to the IEEE digital library Xplore on August 6, 2002, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by August 6, 2002, and is currently available for public download from the IEEE digital library, IEEE Xplore.
12. J. Glass, J. Chang and M. McCandless, "A Probabilistic Framework for Feature-Based Speech Recognition," was published in Proceeding of Fourth International Conference on Spoken Language Processing ICSLP '96, held on October 3-6, 1996, in Philadelphia, PA, USA. Copies of the conference proceedings were made available no later than the last day of the conference. This article was added to the IEEE digital library Xplore on August 6, 2002, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by August 6, 2002, and is currently available for public download from the IEEE digital library, IEEE Xplore.
13. J. R. Glass, T. J. Hazen and I. L. Hetherington, "Real-Time Telephone-Based Speech Recognition in the Jupiter Domain," was published in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings ICASSP99 (Cat. No. 99CH36258), held on March 15-19, 1999, in Phoenix, AZ, USA. Copies of the conference proceedings were made available no later than the last day of the conference. This article was added to the IEEE digital library Xplore on August 6, 2002, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by August 6, 2002, and is currently available for public download from the IEEE digital library, IEEE Xplore.

14. Jilei Tian, J. Nurminen and I. Kiss, "Optimal Subset Selection from Text Databases," was published in Proceedings (ICASSP '05) IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, held on March 23, 2005, in Philadelphia, PA, USA. Copies of the conference proceedings were made available no later than the last day of the conference. This article was added to the IEEE digital library Xplore on May 9, 2005, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by May 9, 2005, and is currently available for public download from the IEEE digital library, IEEE Xplore.

15. Su-Lin Wu, M. L. Shire, S. Greenberg and N. Morgan, "Integrating Syllable Boundary Information into Speech Recognition," was published in 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, held on April 21-24, 1997, in Munich, Germany. Copies of the conference proceedings were made available no later than the last day of the conference. This article was added to the IEEE digital library Xplore on August 6, 2002, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by August 6, 2002, and is currently available for public download from the IEEE digital library, IEEE Xplore.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

I declare under penalty of perjury that the foregoing statements are true and correct.

Date: 6/23/2025

By: Signed by: Gordon MacPherson 845214E4A5C0470...

Heterogeneous lexical units for automatic speech recognition: preliminary investigations

Publisher: IEEE Cite This PDF

I. Bazzi ; J. Glass **All Authors**

Abstract:
This paper explores the use of the phone and syllable as primary units of representation in the first stage of a two-stage recognizer. A finite-state transducer speech recognizer is utilized to configure the recognition as a two-stage process, where either phone or syllable graphs are computed in the first stage, and passed to the second stage to determine the most likely word hypotheses. Preliminary experiments in a weather information speech understanding domain show that a syllable representation with either bigram or trigram language models provides more constraint than a phonetic representation with a higher-order n-gram language model (up to a 6-gram), and approaches the performance of a more conventional single-stage word-based configuration.

Published in: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)

Date of Conference: 05-09 June 2000

DOI: 10.1109/ICASSP.2000.861804

Date Added to IEEE *Xplore*: 06 August 2002

Publisher: IEEE

Print ISBN:0-7803-6293-4

Conference Location: Istanbul, Turkey

Print ISSN: 1520-6149

Authors
References
Citations
Keywords
Metrics

Back to Results

A probabilistic framework for feature-based speech recognition

Publisher: IEEE Cite This PDF

J. Glass ; J. Chang ; M. McCandless **All Authors**

Abstract:
Most current speech recognizers use an observation space which is based on a temporal sequence of "frames" (e.g. Mel-cepstra). There is another class of recognizer which further processes these frames to produce a segment-based network, and represents each segment by fixed-dimensional "features". In such feature-based recognizers, the observation space takes the form of a temporal network of feature vectors, so that a single segmentation of an utterance uses a subset of all possible feature vectors. In this paper, we examine a maximum a-posteriori decoding strategy for feature-based recognizers and develop a normalization criterion that is useful for a segment-based Viterbi or A* search. We report experimental results for the task of phonetic recognition on the TIMIT corpus, where we achieved context-independent and context-dependent (using diphones) results on the core test set of 64.1% and 69.5% respectively.

Published in: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96

Date of Conference: 03-06 October 1996

DOI: 10.1109/ICSLP.1996.607261

Date Added to IEEE *Xplore*: 06 August 2002

Publisher: IEEE

Print ISBN:0-7803-3555-4

Conference Location: Philadelphia, PA, USA

Authors
References
Citations
Keywords
Metrics

Back to Results

Real-time telephone-based speech recognition in the Jupiter domain

Publisher: IEEE Cite This PDF

J.R. Glass ; T.J. Hazen ; I.L. Hetherington **All Authors**

Abstract:
This paper describes our experiences with developing a real-time telephone-based speech recognizer as part of a conversational system in the weather information domain. This system has been used to collect spontaneous speech data which has proven to be extremely valuable for research in a number of different areas. After describing the corpus we have collected, we describe the development of the recognizer vocabulary, pronunciations, language and acoustic models for this system, the new weighted finite-state transducer-based lexical access component, and report on the current performance of the recognizer under several different conditions. We also analyze recognition latency to verify that the system performs in real-time.

Published in: 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)

Date of Conference: 15-19 March 1999

DOI: 10.1109/ICASSP.1999.758062

Date Added to IEEE *Xplore*: 06 August 2002

Publisher: IEEE

Print ISBN:0-7803-5041-3

Conference Location: Phoenix, AZ, USA

Print ISSN: 1520-6149

Authors
References
Citations
Keywords
Metrics

Back to Results

Optimal subset selection from text databases

Publisher: IEEE Cite This PDF

Jilei Tian ; J. Nurminen ; I. Kiss **All Authors**

Abstract:
Speech and language processing techniques, such as automatic speech recognition (ASR), text-to-speech (TTS) synthesis, language understanding and translation, will play a key role in tomorrow's user interfaces. Many of these techniques employ models that must be trained using text data. We introduce a novel method for training set selection from text databases. The quality of the training subset is ensured using an objective function that effectively describes the coverage achieved with the strings in the subset. The validity of the subset selection technique is verified in an automatic syllabification task. The results clearly indicate that the proposed systematic selection approach maximizes the quality of the training set, which in turn improves the quality of the trained model. The presented idea can be used in a wide variety of language processing applications that require training with text databases.

Published in: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.

Date of Conference: 23-23 March 2005

Date Added to IEEE *Xplore*: 09 May 2005

Print ISBN:0-7803-8874-7

ISSN Information:

First Page of the Article

DOI: 10.1109/ICASSP.2005.1415111

Publisher: IEEE

Conference Location: Philadelphia, PA, USA

OPTIMAL SUBSET SELECTION FROM TEXT DATABASES

Jilei Tian, Jani Nurminen and Imre Kiss

Multimedia Technologies Laboratory
Nokia Research Center, Tampere, Finland
{jilei.tian, jani.k.nurminen, imre.kiss}@nokia.com

ABSTRACT

Speech and language processing techniques, such as automatic speech recognition (ASR), text-to-speech (TTS) synthesis, language understanding and translation, will play a key role in tomorrow's user interfaces. Many of these techniques employ models that must be trained using text data. In this paper, we introduce a novel method for training set selection from text databases. The quality of the training subset is ensured using an objective function that effectively describes the coverage achieved with the strings in the subset. The validity of the subset selection technique is verified in an automatic syllabification task. The results clearly indicate that the proposed systematic selection approach maximizes the quality of the training set, which in turn improves the quality of the trained model. The presented idea can be used in a wide variety of language processing applications that require training with text databases.

1 INTRODUCTION

Most automatic speech recognition (ASR) and text-to-speech (TTS) systems contain models that have to be trained with text data. Typical examples can be found from many parts of the systems. In pronunciation modeling, some data-driven approach, such as neural network based methods or decision tree based methods [6], are often applied, especially for languages like English. These statistical models are trained using a pronunciation dictionary containing grapheme-to-phoneme entries. In text-based language identification [8], the model is trained using a multilingual text corpus that consists of word entries from the target languages. In the data-driven syllabification task [7], the model is trained using text-based pronunciations and the corresponding syllable structures.

In all data-driven approaches, the selection of a suitable training set can be regarded as a very important step in the training process. In general, the performance of any trained model depends quite strongly on the quality of the text data used in the training. With text-based data, the importance of the training set selection is very pronounced since the generation of the training data entries is often very time and resource consuming and requires language-specific skills. In this paper, we show that systematic training set selection results in enhanced model performance and/or offers the possibility to use a smaller training set size. In practice, the reduced training set size brings two significant additional benefits. First, the amount of manual annotation work is reduced, which in turn decreases the probability of errors and inconsistencies in the annotations. Second, the memory consumption and the computational load caused by the training process are lowered. In some cases this advantage propagates to the trained model as well; the size of a decision tree model, for example, depends on the size of the training set.

Despite the evident importance of the training set selection, this step is often neglected in practice. Usually, the training set is obtained by collecting a set of random entries from a larger text database or by decimating a sorted corpus. The drawback of these solutions is that the amount of meaningful information in the selected text data set is not maximized. The random selection method is rather coarse and does not produce consistent results. The method of decimating a sorted data corpus, on the other hand, only uses a limited number of the initial characters of the strings and thus does not guarantee good performance.

In this paper, we present a method that can quasi-optimally select a subset from a text database in such a manner that the text coverage is maximized. To achieve this, we define an objective function that is optimized in the subset selection. The objective function measures the "subset distance" using the generalized Levenshtein distances between the text strings. This paper also introduces an algorithm for optimizing the objective function. For practical applications with large databases, the algorithm can be modified in order to speed up the processing or to lower the memory consumption, but the main idea and the objective function will remain useful in all cases. To demonstrate the usefulness of the proposed approach, we evaluate it in the syllabification task.

The text subset selection method introduced in this paper can be used in a wide variety of different applications. One good example is the language identification task [8], in which the proposed approach makes it possible to easily balance the number of training set entries from each target language while at the same time giving a good coverage for every target language. In addition to the training set selection task discussed extensively in this paper, it is possible to employ the same techniques for clustering a text database. Moreover, when used together with a meaningful distance measure, such as the generalized Levenshtein distance, the proposed approach enables the use of vector quantization techniques on text data.

The remainder of the paper is organized as follows. We first describe the generalized Levenshtein distance and introduce the basic principles of the text database selection algorithm in Section 2. In Section 3, we describe the syllabification task used as the practical example by briefly reviewing the syllable structure grammar and the neural network based syllabification method. The performance of the proposed subset selection approach is evaluated in the

0-7803-8874-7/05/$20.00 ©2005 IEEE

SECTION 1
Introduction

Most automatic speech recognition (ASR) and text-to-speech (TTS) systems contain models that have to be trained with text data. Typical examples can be found from many parts of the systems. In pronunciation modeling, some data-driven approach, such as neural network based methods or decision tree based methods [6], are often applied, especially for languages like English. These statistical models are trained using a pronunciation dictionary containing grapheme-to-phoneme entries. In text-based language identification [8], the model is trained using a multilingual text corpus that consists of word entries from the target languages.

In the data-driven syllabification task [7], the model is trained using text-based pronunciations and the corresponding syllable structures.

In all data-driven approaches, the selection of a suitable training set can be regarded as a very important step in the training process. In general, the performance of any trained model depends quite strongly on the quality of the text data used in the training. With text-based data, the importance of the training set selection is very pronounced since the generation of the training data entries is often very time and resource consuming and requires language-specific skills. In this paper, we show that systematic training set selection results in enhanced model performance and/or offers the possibility to use a smaller training set size. In practice, the reduced training set size brings two significant additional benefits. First, the amount of manual annotation work is reduced, which in turn decreases the probability of errors and inconsistencies in the annotations. Second, the memory consumption and the computational load caused by the training process are lowered. In some cases this advantage propagates to the trained model as well; the size of a decision tree model, for example, depends on the size of the training set.

Despite the evident importance of the training set selection, this step is often neglected in practice. Usually, the training set is obtained by collecting a set of random entries from a larger text database or by decimating a sorted corpus. The drawback of these solutions is that the amount of meaningful information in the selected text data set is not maximized. The random selection method is rather coarse and does not produce consistent results. The method of decimating a sorted data corpus, on the other hand, only uses a limited number of the initial characters of the strings and thus does not guarantee good performance.

In this paper, we present a method that can quasi-optimally select a subset from a text database in such a manner that the text coverage is maximized. To achieve this, we define an objective function that is optimized in the subset selection. The objective function measures the “subset distance”. using the generalized Levenshtein distances between the text strings. This paper also introduces an algorithm for optimizing the objective function. For practical applications with large databases, the algorithm can be modified in order to speed up the processing or to lower the memory consumption, but the main idea and the objective function will remain useful in all cases. To demonstrate the usefulness of the proposed approach, we evaluate it in the syllabification task.

The text subset selection method introduced in this paper can be used in a wide variety of different applications. One good example is the language identification task [8], in which the proposed approach makes it possible to easily balance the number of training set entries from each target language while at the same time giving a good coverage for every target language. In addition to the training set selection task discussed extensively in this paper, it is possible to employ the same techniques for clustering a text database. Moreover, when used together with a meaningful distance measure, such as the generalized Levenshtein distance, the proposed approach enables the use of vector quantization techniques on text data.

The remainder of the paper is organized as follows. We first describe the generalized Levenshtein distance and introduce the basic principles of the text database selection algorithm in Section 2. In Section 3, we describe the syllabification task used as the practical example by briefly reviewing the syllable structure grammar and the neural network based syllabification method. The performance of the proposed subset selection approach is evaluated in the syllabification task in Section 4. Finally, some concluding remarks are presented in Section 5.

SECTION 2
Selection Algorithm

In order to be able to select a subset from a text database in a systematic and meaningful manner, an objective function measuring the quality of the subset must be defined. The objective function should somehow measure the similarity or the dissimilarity of the entries. In the proposed approach, we base the objective function on the generalized Levenshtein distance. In this section, we first describe the basic properties of this distance measure and then continue by defining an objective function measuring the average distance within a subset and by introducing an algorithm for selecting subsets of different sizes in a quasi-optimal manner.

2.1 Generalized Levenshtein distance

The generalized Levenshtein distance (GLD) is defined as the minimum cost of transforming one string into another by means of a sequence of basic transformations: insertion, deletion and substitution [4]. The transformation cost is determined by the costs assigned to each basic transformation.

Let x and y be strings of length m and n, respectively, whose symbols belong to a finite alphabet of size S. Let x_i be the ith symbol of string x, with $1 \leq i \leq m$, and $x(i)$ be the prefix of the string x of length i, i.e. the substring containing the first i symbols of x. In addition, let $d(i,j)$ be the distance between $x(i)$ and $y(j)$, and $\mathcal{E}$ be an empty string. Furthermore, we denote by $w(a,b)$, $w(a,\varepsilon)$ and $w(\mathcal{E},b)$ the cost of substituting the symbol a with the symbol b, the cost of deleting a and the cost of inserting b, respectively. The distance $d(m,n)$ is recursively computed based on the definitions of $d(0,0)$, $d(i,0)$ and $d(0,j)(i=1\ldots m, j=1\ldots n)$, representing the initial distance, the cost of deleting the prefix $x(i)$ and the cost of inserting the prefix $y(j)$, respectively, as follows:

$$d(0,0) = 0$$
$$d(i_20) = d(i-1_20) + w(x_l,\varepsilon)\forall i = 1\ldots,m$$
$$d(0,j) = d(0,j-1) + w(\varepsilon,y_j)\forall j = 1\ldots,n \quad (1)$$
$$d(i,j) = \min\begin{cases} d(i-1,j) + w(x_i,e) \\ d(i,j-1) + w(e,y_j) \\ d(i-1,j-1) + w(x_i,y_j) \end{cases} \quad (2)$$

View Source

The original Levenshtein distance is characterized by the following costs: $w(a,\varepsilon) = 1$, $w(\varepsilon,b) = 1$, and $w(a,b)$ is 0 if a is equal to b and 1 otherwise. Its generalized version assumes that different costs can be associated to transformations involving different symbols. In the case of an alphabet of S symbols, this requires a table of size $(s+1)$ times $(s+1)$, called the cost table, to store all the substitution, insertion and deletion costs. It can be shown that the defined distance is a metric if the cost table is symmetric.

2.2 Objective function and selection algorithm

In our approach, we measure the quality of a text subset using an objective function based on the generalized Levenshtein distance. As described in Section 2.1, the Levenshtein distance can be used for measuring the distance between any pair of entries. Similarly, the distance for the whole text data set can be calculated by averaging the distances of all the string pairs in the set. Suppose that there are m entries in the database and the ith entry is denoted by $e(i)$. With these definitions, we can compute the overall "subset distance". D as:

$$D = \frac{2 \cdot \sum_{i=1}^{m}\sum_{j>i}^{m} ld(e(i),e(j))}{m\cdot(m-1)}, \quad (3)$$

View Source

where $ld(e(i),e(j))$ is the GLD between the ith and jth entries.

Based on the above objective function, it is possible to design an algorithm that selects a subset from a text database in such a manner that the distance D is maximized. The following algorithm recursively constructs the subset by always selecting the new entry that maximizes the distance to the other selected entries.

1. Calculate the Levenshtein distances for all the pairs; $ld(e(i),e(j))$;
2. Initially select the pair that has the largest distance among all pairs in the database,

$$((subset\_e(1), subset\_e(2)) = \arg\max_{(l\leq m, j>i)}\{ld(e(i),e(j))\}. \quad (4)$$

View Source

3. Assuming that the selected subset has k entries (in the first time $k = 2$), the target now is to find the $k + 1$-th entry to the subset. The selection that approximately maximizes the amount of new information brought into the subset can be done using the following formula.

$$p = \arg \max_{(l \geq i \geq m)} \{ \sum_{j=1, e(i) \neq subset\_e(j)}^{k} , subset\_e(j) \}. \quad (5)$$

View Source

The selected entry p is added into the subset as $subset\_e(k + 1)$.

4. Repeat step 3 until the preset subset size is reached.

SECTION 3
Example Application: Syllabification Task

The development of speech synthesizers and speech recognizers often requires working with sub-word units such as syllables [5]. We have earlier described a neural network based approach for the automatic assignment of syllable boundaries in [7]. In this paper, we revisit the topic and use this syllabification task for verifying the usefulness of the proposed subset selection approach. The first part of this section gives some basic information on the task and the second part discusses the neural network approach. The practical results achieved in this task are presented in Section 4.

3.1 Syllable structure

A syllable is a basic unit of word studied on both the phonetic and phonological levels of analysis [2]. The syllable information can be described using grammars [3]. The simplest grammar is the phoneme grammar, where a syllable is tagged with the corresponding phoneme sequence. The consonant-vowel grammar describes a syllable as a consonant-vowel-consonant (CVC) sequence. The syllable structure grammar, on the other hand, divides a syllable into onset, nucleus and coda (ONC) as shown in Figure 1. The nucleus is an obligatory part that can be either a vowel or a diphthong. The onset is the first part of a syllable consisting of consonants and ending at the nucleus of the syllable, e.g. in the syllable $[tehkst]$, $/t/$ is the onset and the vowel part $/eh/$ is the nucleus. The part of a syllable that follows the nucleus forms the coda. The coda is constructed of consonants, e.g. $/kst/$ in our example syllable. The nucleus and coda are combined to form the rhyme of a syllable. A syllable has a rhyme, even if it doesn't have a coda.

In the syllable structure grammar, the consonants are assigned as onset or coda. The ONC representation used in the syllable structure grammar contains more information than the CVC structure for multi-syllable words. The syllable structure grammar was used in [7] and it is also used in this paper.

In the automatic syllabification task, the phoneme sequences are mapped into their ONC representations. The data-driven syllabification model is trained on the mapping information. In the decoding phase, given a phoneme sequence, the ONC sequence is first generated, and then the syllable boundaries are uniquely decided on the ONC sequence. For invalid ONC sequences, a self-correction algorithm [7] can be applied to solve the problem by utilizing certain common linguistic rules. The whole syllabification task can be summarized as follows:

1 Each pronunciation phoneme string in the training set is mapped into the corresponding ONC string, for example: (word) $text->$ (pronunciation) $tehkst->$ (ONC)$ONCCC$

2 The model is trained on the data in the format of "pronunciation -> ONC". 3. Given a pronunciation string, the corresponding ONC sequence is generated using the model. Then, the syllable boundaries are placed at the location starting with symbol "O," or with "N" if it is not preceded with symbol "O".

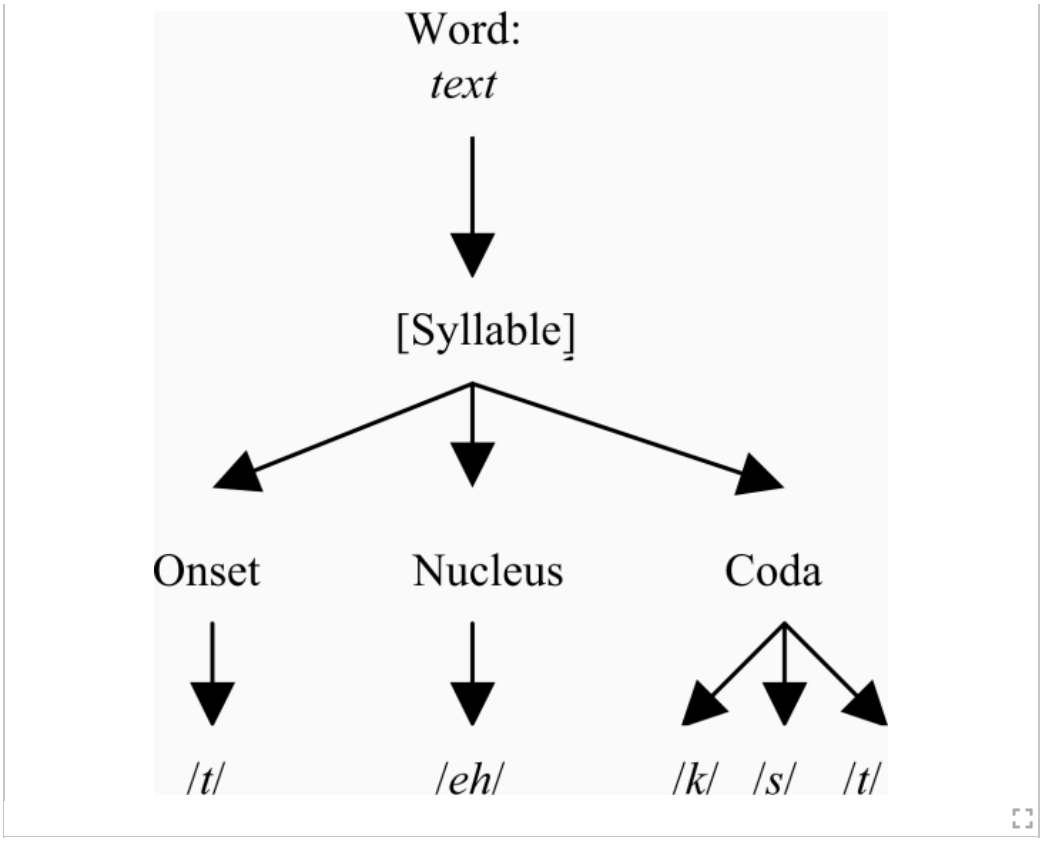

Figure 1. Diagram of the syllable structure grammar.

3.2 Neural network based syllabification approach

The basic neural network based ONC model presented in [7] is a standard multi-layer perceptron (MLP) shown in Figure 2. The input phonemes are presented to the MLP network in a sequential manner. The network gives estimates of ONC posterior probabilities for each presented phoneme. In order to take the phoneme context into account, a number of phonemes on each side of the phoneme in question are also used as inputs to the network. Thus, a window of phonemes is presented to the neural network as input. Figure 2 shows a typical MLP with a context size of w phonemes, $ph_{i-w} \ldots ph_{i+w}$ centered at phoneme ph_i. The centermost phoneme ph_i is the phoneme that corresponds to the output of the network. Therefore, the output of the MLP is the estimated ONC probability $P(onc_k|ph_{i-w} \ldots, ph_{i+w})(onc_k \in \{O, N, C\})$ for the centermost phoneme ph_i in the given context $p_{i-w} \ldots p_{i+u}$. A phonemic null is defined in the phoneme set and is used for representing phonemes to the left of the first phoneme and to the right of the last phoneme in a pronunciation.

The ONC neural network is a fully connected MLP, which uses a hyperbolic tangent sigmoid shaped function in the hidden layer and a softmax normalization function in the output layer. The softmax normalization ensures that the network outputs are in the range [0], [1] and sum up to unity,

$$P_i = \frac{e^{y_i}}{\sum_{j=1}^{3} e^{y_j}}. \tag{6}$$

View Source

In Equation (6), y_i and P_i denote the ith output value before and after softmax normalization. It has been shown in [1] that a neural network with softmax normalization will approximate class posterior probabilities when trained for one-out-of-N classification and when the network is sufficiently complex and trained to a global minimum. Since the neural network input units are text-valued, the phonemes in the input window need to be transformed to some numeric quantity. This can be done, for example, using an orthogonal codebook representing the alphabet used for the ONC mapping task, as shown in Table 1. The last row in the table is the code for the phonemic null. An important property of the orthogonal coding scheme is that it does not introduce any correlation between the different letters.

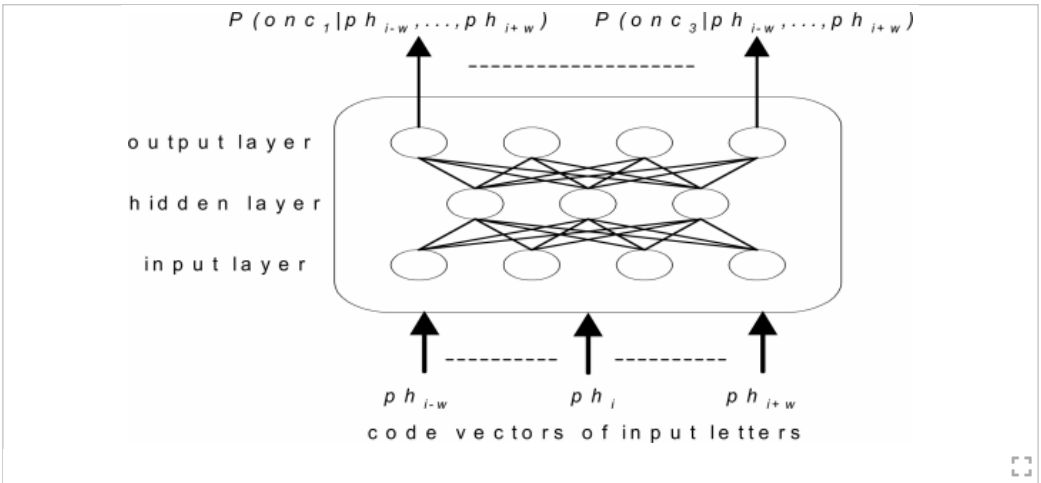

Figure 2. Two-layer neural network architecture.

The ONC neural network is trained using the standard back-propagation (BP) algorithm augmented by a momentum term. Each phoneme with context and the corresponding ONC tag of the pronunciation make up one training example. Weights are updated in a stochastic on-line fashion. All parameters are rounded off to eight bits as this was found sufficient for representing model parameters.

Table 1. Orthogonal phoneme coding scheme.

Letter	Code
aa	100...0000
ae	010...0000
...	...
B	000...1000
P	000...0100
T	000...0010
#	000...0001

The outputs of the ONC neural network approximate the ONC posterior probabilities corresponding to the centermost phoneme. The ONC sequence of a pronunciation is obtained by combining the network outputs for each individual phoneme in the pronunciation. Given a pronunciation with its phonemic representation, the ONC tag of phoneme ph_i is given by

$$onc = \arg\max_{noc_k}\{P(onc_k|ph_{i-w},\ldots,ph_{i+w})\}, \quad (7)$$

View Source

where $P(onc_k|ph_{i-w},\ldots,ph_{i+w})$ is the network output corresponding to onc_k given the input phonemes $ph_{i-w}\ldots ph_{i+w}$, and variable w denotes the phoneme window context size, respectively. The variable onc takes its values from the set $[ONC]$.

SECTION 4
Experimental Results

The neural network based syllabification method is evaluated using the CMU dictionary for US English. The dictionary contains 10,801 words with their pronunciations and labels with ONC information. The pronunciations and the mapped ONC sequences are extracted to form the training data. The training set is selected from the whole database by using the following methods:

- Decimation of the sorted dictionary (denoted as DECIMATE);
- Subset selection from the text database using the selection approach proposed in this paper (denoted as SELECT).

With both methods, the data not selected to the training set constitutes the test set.

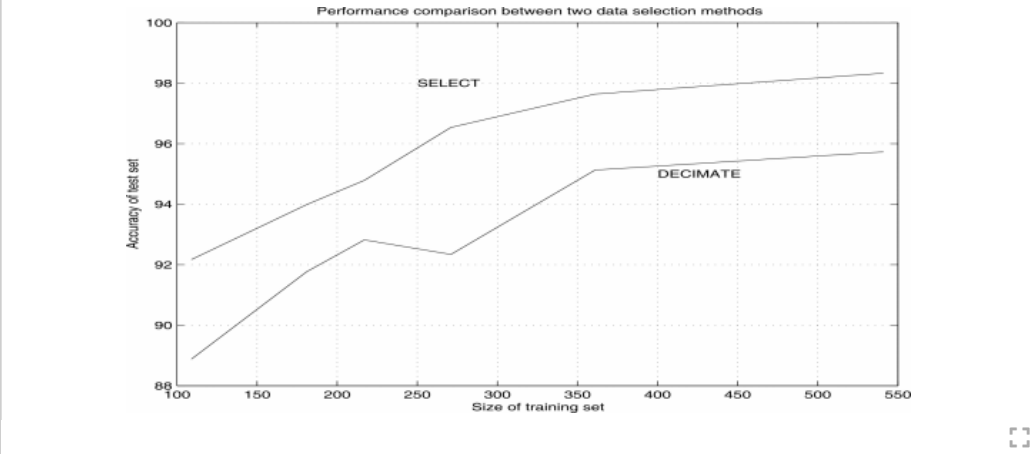

Figure 3. ONC accuracy on test set with different training set sizes using the two data selection methods.

Figure 3 shows the experimental results achieved using the two data selection methods. The efficiency of the training set selection approach can be studied by evaluating the generalization capability. The general rule of thumb is that the more training data is available, the better performance can be expected. However, the selection of the training data affects the generalization capability: if the training data is well selected, the performance can be improved without increasing the size of the training set. The results clearly show that the proposed subset selection technique outperforms the commonly used decimation method; the average improvement achieved using the proposed approach is 38.8%.

Figure 4 illustrates the "subset distance". (see Section 2.2) of datasets extracted using the two different data selection methods: the decimation technique and the proposed selection algorithm. It is easy to see that the average distance D is more or less even when the decimation method is used. With the proposed method, the average distance decreases monotonically with increasing data size. Furthermore, the difference between the two methods is large with small subset sizes, and converges to zero when the whole data set is used. Thus, these results indicate that the proposed method can extract data more efficiently, i.e. the selected data has better coverage. Naturally, this explains the better generalization capability of the trained model.

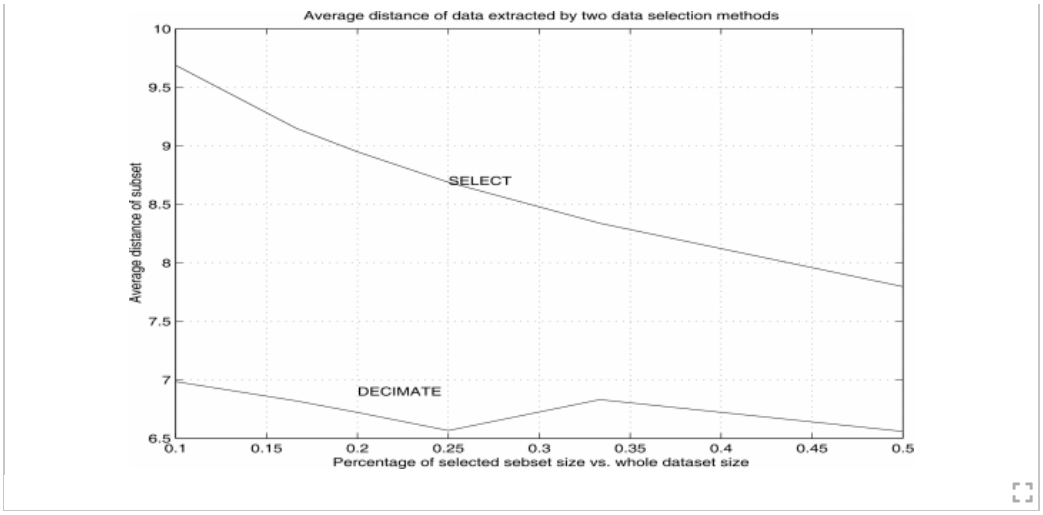

Figure 4. Average distance D inside the subsets extracted using the two different data selection methods, with respect to the percentage of the subset size vs. the whole data size.

SECTION 5
Conclusions

Training data selection from a text database is a crucial, but often neglected, step in the development of ASR and TTS systems. In this paper, we define an objective function that effectively measures the quality of a selected subset. Moreover, we introduce a subset selection algorithm that optimizes the objective function. Our experimental results obtained in the syllabification task show that the proposed approach is a very promising technique that makes it possible to select subsets with good coverage in a systematic and meaningful way. The presented idea can be used in many different applications that require training with a text database.

Authors
Figures
References
Citations
Keywords
Metrics

Back to Results

Integrating syllable boundary information into speech recognition

Publisher: IEEE Cite This PDF

Su-Lin Wu ; M.L. Shire ; S. Greenberg ; N. Morgan **All Authors**

Abstract:
We examine the proposition that knowledge of the timing of syllabic onsets may be useful in improving the performance of speech recognition systems. A method of estimating the location of syllable onsets derived from the analysis of energy trajectories in critical band channels has been developed, and a syllable-based decoder has been designed and implemented that incorporates this onset information into the speech recognition process. For a small, continuous speech recognition task the addition of artificial syllabic onset information (derived from advance knowledge of the word transcriptions) lowers the word error rate by 38%. Incorporating acoustically-derived syllabic onset information reduces the word error rate by 10% on the same task. The latter experiment has highlighted representational issues on coordinating acoustic and lexical syllabifications, a topic we are beginning to explore.

Published in: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing

Date of Conference: 21-24 April 1997

DOI: 10.1109/ICASSP.1997.596105

Date Added to IEEE *Xplore*: 06 August 2002

Publisher: IEEE

Print ISBN:0-8186-7919-0

Conference Location: Munich, Germany

Print ISSN: 1520-6149

Authors
References
Citations
Keywords
Metrics

Back to Results

EXHIBIT A

HETEROGENEOUS LEXICAL UNITS FOR AUTOMATIC SPEECH RECOGNITION: PRELIMINARY INVESTIGATIONS

Issam Bazzi and James Glass

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 USA

ABSTRACT

This paper explores the use of the phone and syllable as primary units of representation in the first stage of a two-stage recognizer. A finite-state transducer speech recognizer is utilized to configure the recognition as a two-stage process, where either phone or syllable graphs are computed in the first stage, and passed to the second stage to determine the most likely word hypotheses. Preliminary experiments in a weather information speech understanding domain show that a syllable representation with either bigram or trigram language models provides more constraint than a phonetic representation with a higher-order n-gram language model (up to a 6-gram), and approaches the performance of a more conventional single-stage word-based configuration.

1. INTRODUCTION

Most conventional speech recognition systems represent the search space as a directed graph of phone-like units. These graphs are typically determined by the allowable pronunciations of a given word vocabulary, with word (and thus phone) sequences being prioritized by word-level constraints such as n-grams. This framework has proven to be very effective, since it combines multiple knowledge sources into a single search space, rather than decoupling the search into multiple stages, each with the potential to introduce errors. Although multi-stage searches have been explored, they typically all operate with the word as a basic unit.

Although this framework has worked extremely well, the use of the word as the main unit of representation has some difficulties in certain situations. One common problem is that for any reasonably-sized domain, it is essentially impossible to predefine a word vocabulary. For example, in our weather information system, we are constantly faced with new words spoken by users (e.g., city names, concepts). We would have this problem no matter how large our vocabulary was, since the vocabulary of the English language is constantly growing and changing. It is thus not possible to define a word vocabulary, no matter how large, that will forever cover all conceivable spoken words. One problem with out-of-vocabulary words, is that they introduce errors into the recognition system (typically more than one), since the recognizer will fit the phonetic sequence with the best-fitting set of words which exist in its vocabulary.

A similar phenomenon to out-of-vocabulary words is that of partially spoken words, which are typically produced in more conversational or spontaneous speech applications. These phenomena also tend to produce errors since the recognizer matches the phonetic sequence with the best fitting words in its active vocabulary.

Since the domains we work on tend to have both of these properties, we have begun to explore methods that can be used to model out-of-vocabulary and partial words which are based on the use of more flexible sub-word units (such as phones or syllables, which are not constrained to match the active word vocabulary). Sub-word units such as phones and syllables have the attractive property of being a closed set, and thus will be able to cover new words, and can conceivably cover most partial word utterances as well. While these methods can conceivably fit within a domain-dependent word-based recognition architecture, we are also interested in exploring their use as a separate first stage, operating independently of a given vocabulary.

One of the main reasons for exploring the utility of a domain-independent first stage is to attempt to separate domain-independent constraints from domain-dependent one in the speech recognition process. Currently, most speech recognizers are tuned to a particular domain, for both acoustic and linguistic modeling. In our experience, this is especially true of the language model, which can provide tremendous constraint to the search space. We are interested in exploring the viability of incorporating domain-independent constraints in a first-stage process, and leaving domain-dependent constraints to a second-stage search.

For speech understanding systems, a two-stage recognizer might enable alternative integration strategies with natural language understanding. To date, most such systems are loosely coupled at the word-level via N-best or word graph interfaces. An alternative unit might allow for more integrated search strategies (e.g., [1]), with a unified word-based language model.

A two-stage recognizer configuration might also provide for a more flexible deployment strategy. For example, a user interacting with several different spoken dialogue do-

This material is based upon work supported by the National Science Foundation under Grant No. IRI-9618731

0-7803-6293-4/00/$10.00 ©2000 IEEE.

mains (e.g., weather, travel, entertainment), might have their speech initially processed by a domain-independent first stage, and then subsequently processed by domain-dependent recognizers. For client/server architectures, a two-stage recognition process could be configured to have the first stage run locally on small client devices (e.g., hand-held portables) and thus potentially require less bandwidth to communicate with remote servers for the second stage.

In this work we have begun to examine whether we can create a two-stage recognizer with a domain-independent first stage, without sacrificing accuracy due to the lack of word-level constraints in the first stage. In particular we were interested in understanding whether better-trained (due to fewer units) sub-word models could provide a useful source of information to our recognizer.

Although we are ultimately interested in out-of-vocabulary and partial word phenomena, as well as domain-independence, these topics have not been part of these initial investigations. Instead we have examined only the best-case scenario for a word-based recognizer (i.e., within-vocabulary utterances only). Our motivation was to establish that a two-stage system could at least be competitive in this environment, since we hope that it can surpass a word-based approach in non-optimal situations. We have also allowed our first-stage recognizers to be domain-dependent, to establish at least an upper bound on performance. The following sections outline our strategy and report preliminary experiments with two possible sub-word representations, namely the phone and the syllable.

2. RECOGNIZER ARCHITECTURE

In this work we use the SUMMIT segment-based speech recognition system [2]. Typical recognizer configurations deploy a bigram language model in a forward Viterbi search, while a trigram (or higher-order) language model is used in a backward A^* search. The SUMMIT system uses a weighted finite-state transducer (FST) representation of the search space [3]. In this framework, recognition can be viewed as finding the best path(s) in the composition:

$$S = P \circ L \circ G, \tag{1}$$

where P represents the scored phonetic graph, L is the lexicon mapping pronunciations to lexical units, and G is the language model. Equation (1) shows how a typical recognizer is formulated as a compositions of three FST's. However, this FST framework allows for a variety of compositions and flexibility in the composition order. In typical recognizer configurations, L and G are precomposed prior to recognition, and are then composed with P during recognition to create one single large search space. In the following sections, we describe how we can divide this composition into two stages, using either phones or syllables as the first-stage unit of representation.

2.1. The Phone Recognizer

A two-stage search using phones as the first-stage unit of representation can be represented in FST notation as:

$$S = P \circ L_p \circ G_p \circ L \circ G \tag{2}$$

where L_p and G_p are the phone lexicon and grammar, respectively, while L and G are the corresponding word lexicon and grammar, which are the same as those in the basic word recognizer configuration. For our phone recognizer, L_p is a trivial FST and can be discarded, since the phone units in P are already the basic units of the word lexicon. The phone grammar, G_p, can consist of a phone-level n-gram language model. Since the phone inventory size is small we are able to run with higher-order n-grams than we would be able to with words.

Although there are many possible ways to explore the phone composition, S, we have only explored one way thus far. In our experiments, we precompose L and G as in the baseline word recognizer. During the first stage of recognition, we compute a phone graph from the composition of P and G_p. This graph is then composed with the word FST to produce the best word hypothesis. We express this search order in the following expression:

$$S = (P \circ G_p) \circ (L \circ G) \tag{3}$$

Since the phone vocabulary is quite small, the first stage can potentially be much faster than the baseline word recognition system. We wished to understand how much the phone grammar G_p could compensate for the loss of higher-level word constraints during the first stage, and whether the two-stage search would suffer higher word error rates.

2.2. The Syllable Recognizer

A two-stage search using syllables as the first-stage unit of representation can be represented in FST notation as:

$$S = P \circ L_s \circ G_s \circ L_w \circ G \tag{4}$$

where L_s and G_s are the syllable lexicon and grammar, respectively. The syllable lexicon, L_s, is created from the word lexicon, L, through a direct mapping from phonetic units to syllabic units. For each word in the lexicon, we partition the phone sequence into syllables using an automatic syllabification procedure [5]. Entries in the second-stage word lexicon, L_w, are represented by sequences of syllable units. Syllable graphs are used to represent words with multiple pronunciations.

To build the syllable language model, G_s, we start with a word-based training set, and partition the words into syllables to obtain syllable sequences for training a syllable bigram or trigram. For words with multiple pronunciations, we randomly select one of the allowed pronunciations and use the corresponding sequence of syllables.

The two-stage search configuration for the syllable-based recognizer is similar to the phone-based recognizer. In the first stage we compute a syllable graph by searching the composition of P with the precomposed FST $L_s \circ G_s$. The second-stage search composes this FST with the precomposed word FST $L_w \circ G$. We describe this search as:

$$S = (P \circ L_s \circ G_s) \circ (L_w \circ G) \tag{5}$$

For the syllable-based experiments, we were interested in learning whether syllable constraints in the first stage could better compensate for the loss of word information than could phonetic constraints alone.

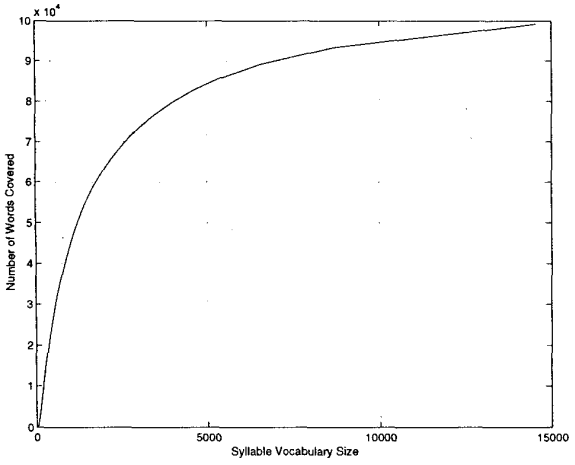

Figure 1: Word coverage versus syllable vocabulary size.

In order to explore the degradation of splitting the search into two stages, we also examined a single composition and search for the syllable representation. In these experiments, $L_s \circ G_s$, and $L_w \circ G$ were precomposed, but the final composition with P was done dynamically in a single search. The dynamic composition allowed us to explore the full search space in a single pass.

Although a closed-set syllable recognizer would require all possible syllables for a given language, in practice it might be desirable to utilize a subset of syllables which provide good coverage for a particular domain. The subset could be created via a selection criterion which maximizes coverage of a particular vocabulary.

To better understand vocabulary coverage with syllables, we examined the LDC PRONLEX dictionary which contains 90,694 words with 99,202 unique pronunciations. When these pronunciations were syllabified we obtained a total of 14,570 syllables. Figure 1 plots the vocabulary coverage as a function of the number of syllables. Our selection criterion was based on the most frequently occurring syllables in the lexicon. The figure indicates that the coverage quickly increases as we add more syllables to the inventory. For example, using a syllable inventory of 1,000 syllables covers around 45,000 words, a fairly large coverage for a relatively small syllable vocabulary.

3. EXPERIMENTS AND RESULTS

The experiments described in this section are all within the JUPITER weather information domain [3]. In the following sections we first give a brief description of the baseline system and report both word and phonetic error rates. We then present phonetic error rates for the first stage of the phone and syllable recognizers. Finally, we report word error rates of the full two-stage systems.

Recognition unit	n-gram order	PER (%)
Word	2	5.9
Phone	3	24.0
Phone	4	19.5
Phone	5	17.4
Phone	6	15.9
Syllable	2	14.3
Syllable	3	12.1

Table 1: Phonetic error rates for first stage recognizers.

3.1. The Baseline System

The baseline system used a similar configuration to that which has been reported previously [3]. A set of context-dependent diphone acoustic models were used, whose feature representation was based on the first 14 MFCC's averaged over 8 regions near hypothesized phonetic boundaries. Diphones were modeled using diagonal Gaussians with a maximum of 50 mixtures per model. The word lexicon consisted of a total of 1957 words, many of which have multiple pronunciations [3]. The training set used for these experiments consists of 46,685 utterances used to train both the acoustic and the language models. The test set consists of 1169 utterances. This test data consists of sets of calls randomly selected over our data collection period [3].

3.2. Phonetic Recognition Experiments

Since we wanted to be able to compare performance across our different recognizer configurations, we first evaluated the phonetic error rate (PER) for each system. Reference phonetic transcriptions were computed by creating forced paths (i.e., constrained by the orthography). The PER for the baseline word-based system was computed by taking the best phonetic sequence of the top word hypothesis (i.e., Viterbi output). As shown in Table 1, we obtained a 5.9% PER for the baseline word-based system.

For the phone-based recognizer, the phone lexicon, L_p, consisted of all phonetic units in JUPITER. The resulting vocabulary size of the phone recognizer was 61 phones. We experimented with four different phone n-gram models (n=3-6) for G_p. Table 1 shows the PER as a function of the n-gram order. Going from a trigram to a 6-gram, we note around 32% reduction in PER (from 24.0% to 15.9%).

For the syllable-based recognizer, we started with the JUPITER vocabulary of 1957 words. Breaking the words into syllables, we obtained a syllable lexicon, L_s, of 1624 syllables. For the syllable n-gram, G_s, we experimented with both a bigram and a trigram on sequences of syllables. As we can see from Table 1, we obtained a PER of 14.3% with a syllable bigram, and 12.1% with a syllable trigram.

3.3. Word Recognition Experiments

Following the first-stage experiments, we evaluated the word error rate (WER) performance for the baseline word-based system, and the phone- and syllable-based systems. As shown in Table 2, the WER for the baseline system using a bigram word-level language model is 10.4%.

Condition	WER (%)
Baseline, word-level	10.4
Phone graphs	15.7
Syllable graphs	13.2
Syllable-level,word composition	11.7

Table 2: Word error rates for word-, phone-, and syllable-based recognizers.

For the phone-based recognizer, we considered a two-stage search which used the best performing n-gram for G_p (i.e., n=6). When the phone graph output was composed with the word-level lexicon and grammar, $L \circ G$, the WER was 15.7%. For the syllable-based two-stage search, we used a syllable trigram for G_s. When the syllable graph was composed with the word-level lexicon and grammar, $L_w \circ G$, the WER was reduced to 13.2%. Finally, when the syllable-based representation was dynamically searched in a single pass, the WER was reduced further to 11.7%.

An important aspect of the two-stage system is the size of the graph. Usually, the bushier (more arcs and states) the graph is, the better the recognition performance. In the limit, if there were no pruning during search, the two stage search would produce identical results to a single stage. However, as the graph size increases, the computation can become quite expensive and does not justify the extra gain in performance. For the experiments we reported, the graph size varied from 1,000 to 10,000 states (and around twice that for the number of arcs) depending on the length of the utterance and uncertainty of the decoder.

4. DISCUSSION

One of the most striking observations from our experiments is the significantly lower phonetic error rate for the word-based recognizer (5.9%) compared to the other recognizers (>12%). However, the WER is only around 11% more (10.4% compared to 11.7%). This suggests that the constraint imposed from using words as the unit of representation does not add significantly to the recognition performance. Thus, a syllable-based representation has some promise as a first-stage unit of representation, due to its increased flexibility.

Despite the use of high-order n-grams, the phone-based recognizer was not as competitive as the syllable-based recognizer, at either of the phonetic or word levels. Even though the use of the 6-gram may capture some information across words, it appears to be less constraining than either the word bigram or the syllable bigram or trigram.

An analysis of the syllable-graph outputs indicates that there remain additional gains to be made for the syllable recognizer. Our word syllabification produced a single possible syllable sequence. In practice, however, we observed that a correct phone sequence would often be represented by a syllable sequence which did not match the underlying sequence because an ambisyllabic consonant had moved to a neighboring syllable. This effect introduced word recognition errors, which we believe can be reduced by representing words as syllable graphs, rather than single sequences.

5. CONCLUSIONS

There are still several computational and modeling issues to resolve that we believe are behind the degradation in word recognition performance for the two-stage framework. Considering the fact that the syllable-based framework is less constrained than the word-based framework, we believe that these preliminary results are quite encouraging.

One of the problems with a two-stage search is the introduction of errors when the correct sub-word sequence is pruned from the intermediate graph. We have been investigating the use of a more flexible matching process in the second stage to compensate for these errors. The matching is done via a confusion FST, which allows for substitution, insertion, and deletion of sub-word units in the graph, and which has been used successfully elsewhere [4].

The experiments performed in this paper were conducted within the context of a single domain. Both the phonetic and syllable recognizers took advantage of the constraints of the domain (e.g., syllable inventory, n-gram grammars). For our future work, we plan to examine the use of a more domain-independent syllable recognizer with a larger inventory of syllables, and a more generic language model. Such a recognizer could easily be combined with a domain-specific word-level lexicon and language model. A domain-independent first stage would not necessarily be composed of a single type of unit. We plan to explore integrating several different lexical units within the same recognizer (e.g., words and syllables). The most frequently spoken words in most domains are function words or particles, and could conceivably add constraint to a language model. Such words also tend to be domain-independent.

Finally, we have not yet examined the behavior of our systems on out-of-vocabulary or partial words. The performance of our word-based systems are significantly worse on these kinds of data, so it is conceivable that recognizer configurations with closed-set units are better able to process these data. We plan to develop a mechanism for handling these phenomena in our second-stage recognizers in the near future.

Acknowledgments

Lee Hetherington provided many helpful suggestions, and helped in implementing tools for manipulating FST's.

REFERENCES

[1] G. Chung and S. Seneff, "Improvements in speech understanding accuracy through the integration of hierarchical linguistic, prosodic, and phonological constraints in the JUPITER domain," in *Proc. ICSLP*, Sydney, 1998.

[2] J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. ICSLP*, Philadelphia, PA, pp. 2277–2280, 1996.

[3] J. Glass, T. Hazen, and L. Hetherington, "Real-time telephone-based speech recognition in the JUPITER domain," in *Proc. ICASSP*, Phoenix, AZ, 1999.

[4] K. Livescu, *Analysis and modeling of non-native speech for automatic speech recognition.* S.M. thesis, MIT, cambridge, August 1999.

[5] J. Yi, *Natural-sounding speech synthesis using variable-length units.* M.Eng. thesis, MIT, Cambridge, May 1998.

EXHIBIT B

A PROBABILISTIC FRAMEWORK FOR FEATURE-BASED SPEECH RECOGNITION[1]

James Glass, Jane Chang, and Michael McCandless

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 USA

ABSTRACT

Most current speech recognizers use an observation space which is based on a temporal sequence of "frames" (e.g., Mel-cepstra). There is another class of recognizer which further processes these frames to produce a segment-based network, and represents each segment by fixed-dimensional "features." In such feature-based recognizers the observation space takes the form of a temporal network of feature vectors, so that a single segmentation of an utterance will use a subset of all possible feature vectors. In this work we examine a maximum *a posteriori* decoding strategy for feature-based recognizers and develop a normalization criterion useful for a segment-based Viterbi or A^* search. We report experimental results for the task of phonetic recognition on the TIMIT corpus where we achieved context-independent and context-dependent (using diphones) results on the core test set of 64.1% and 69.5% respectively.

1. INTRODUCTION

The SUMMIT speech recognizer developed by our group uses a segment-based framework for its acoustic-phonetic representation of the speech signal [22]. Feature vectors are extracted both over hypothesized segments and at their boundaries for phonetic analysis. The resulting observation space (the set of all feature vectors) takes the form of an acoustic-phonetic *network*, whereby different paths through the network are associated with different sets of feature vectors. This framework is quite different from prevailing approaches which employ a temporal *sequence* of observations. The segmental and feature-extraction characteristics of this recognizer provide us with a framework within which we try to incorporate knowledge of the speech signal. They enable us to explore different strategies for where to extract information from the speech signal, and allow us to consider a larger variety of observations than we could with traditional frame-based observations.

We have always tried to cast the recognizer within a probabilistic framework in order to account for our incomplete knowledge. We have been troubled, however, that different paths through our segment-network compute likelihoods on essentially different observation spaces (different segments have different feature vectors), yet our decoder compares the likelihoods of each path to decide on the most-likely word sequence. Additionally, while we train models based on positive examples of our lexical units (e.g., phones), we compute and rank model likelihoods on segments which are not valid units during decoding. This problem is especially serious if likelihoods are converted to posterior probabilities, since a poor likelihood could result in a very good posterior probability only because it happens to be a little better than the (positive) alternatives.

Recently we have reexamined the probabilistic framework we have been using and have adopted a new strategy which we believe better accounts for our feature-based observation space, is intuitively appealing, and reduces the number of tuning parameters required by our system. We now utilize the entire network of hypothesized segments (both positive and negative examples) during training, and try to account for the entire observation space during decoding.

In this paper we show how we derived this framework from basic MAP decoding principles, and present a normalization criterion which can be used to implement efficient decoding for a feature-based recognizer. We then report experimental evidence on phonetic recognition which we have used to evaluate the framework.

2. MAP DECODING

In most probabilistic formulations of speech recognition the goal is to find the sequence of words $W^* = w_1, \ldots, w_N$, which has the maximum *a posteriori* (MAP) probability $P(W|A)$, where A is the set of acoustic observations associated with the speech utterance:

$$W^* = \arg\max_W P(W|A)$$

In most speech recognizers, MAP decoding is accomplished by hypothesizing (usually implicitly) a segmentation S of the utterance into a connected sequence of lexical states or units. In these cases $P(W|A)$ can be rewritten as

$$P(W|A) = \sum_S P(WS|A) \approx \max_S P(WS|A)$$

The latter approximation assumes that there is a single "correct" segmentation S^* associated with W^*. This approximation simplifies the decoding process by allowing the use of dynamic programming algorithms which seek only the "best" path (e.g., Viterbi, or A^*).

[1]This research was supported by DARPA under contract N66001-94-C-6040, monitored though Naval Command, Control and Ocean Surveillance Center. J. Chang receives support from Lucent Technologies.

The expression for $P(WS|A)$ is typically converted to the form:

$$P(WS|A) = \frac{P(AS|W)P(W)}{P(A)}$$

Since the denominator is independent of S or W, it is usually ignored during decoding. The remaining terms $P(AS|W)$ and $P(W)$ are usually estimated separately by acoustic and language models, respectively. In many formulations, such as hidden Markov models (HMMs), the term $P(AS|W)$ is further decomposed into

$$P(AS|W) = P(A|SW)P(S|W)$$

where $P(S|W)$ determines the probability of a particular segmentation (e.g., the HMM state sequence likelihood). $P(A|SW)$ determines the likelihood of seeing the acoustic observations given a particular segmentation (or state sequence).

2.1. Frame-based Observations

Most speech recognizers take as input a temporal sequence of vectors or frames, $O = \{o_1, \ldots, o_T\}$, which are normally computed at regular time intervals (e.g., 10 ms). In most cases a frame contains some form of short-term spectral information (e.g., Mel-cepstra). When the observation space consists of a sequence of frames, $A = O$, and acoustic likelihoods are computed for *every* frame during decoding. Thus, the term $P(A|SW)$ accounts for *all* observations, and competing word hypotheses can be compared directly to each other since their acoustic likelihood is derived from the same observation space. Note that by definition A includes all observations so the denominator term $P(A)$ can be ignored.

As mentioned previously, most recognizers use frame-based observations for input to the decoder. Thus all discrete and continuous HMMs, including those using artificial neural networks for classification, fit under this framework [7, 12, 15, 16, 21]. Many segment-based techniques also use a common set of fixed observation vectors as well. Marcus for example, predetermines a set of acoustic-phonetic sub-segments, represents each by an observation vector, which is then modelled with an HMM [11]. Other segment-based techniques hypothesize segments, but compute likelihoods on a set of observation frames [2, 6, 10, 19].

2.2. Feature-based Observations

In contrast to frame-based approaches, in a *feature*-based framework, each segment s_i is represented by a single fixed-dimensional feature vector x_i. Typically, there is an extra stage of processing to convert the frame sequence O to corresponding features. Explicit segment or boundary hypotheses are necessary to compute the feature vector. A given n unit segmentation $S = s_1, \ldots, s_n$ will have a set of corresponding n feature vectors $X = x_1, \ldots, x_n$. As illustrated in Figure 1, the observation space is transformed from a temporal sequence to a network, where different segmentations of the utterance will be associated with different feature-vectors.

Since alternative segmentations will consist of *different* observation spaces, it is incorrect to compare the resulting likelihoods directly. In order to compare two paths we must consider the *entire* observation space. Thus, in addition to the feature vectors X associated

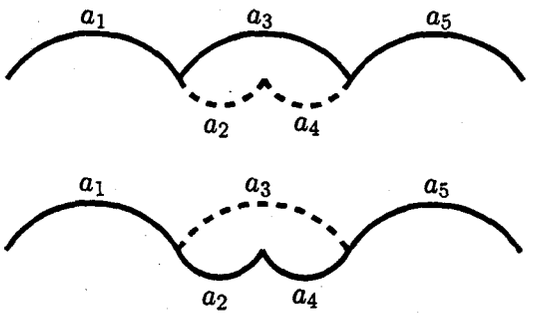

Figure 1: Two segmentations through a segment network with associated feature vectors $\{a_1, \ldots, a_5\}$. The top path uses vectors $\{a_1, a_3, a_5\}$, while the bottom path uses $\{a_1, a_2, a_4, a_5\}$.

with the segmentation S, we must consider all other possible feature vectors in the space Y, corresponding to the set of all other possible segments R. In the top path in Figure 1, $X = \{a_1, a_3, a_5\}$, and $Y = \{a_2, a_4\}$. In the bottom path, $X = \{a_1, a_2, a_4, a_5\}$, and $Y = \{a_3\}$. The total observation space A, contains *both* X and Y, so for MAP decoding it is necessary to estimate $P(XY|SW)$. Note that since S implies X we can say $P(XY|SW) = P(XY|W)$.

In practice, most feature-based recognition systems have *not* estimated a probability for $P(XY|W)$ but have only estimated the likelihood of X, $P(X|W)$ [4, 9, 13, 22]. The following section discusses one method for estimating $P(XY|W)$ in an efficient manner.

3. MODELLING NON-LEXICAL UNITS

One approach to modelling $P(XY|W)$ is to add an extra class to the lexical units which is defined to map to all segments which do *not* correspond to one of the existing units. Consider the case where acoustic-modelling is done at the phonetic level, so that we build probabilistic models for individual phones, $\{\alpha\}$. In this approach we can view the the segments in R as corresponding to the extra *anti-phone* class $\bar{\alpha}$. This class contains all types of sounds which are *not* a phonetic unit as they are either too large, too small, or overlapping etc. Two competing paths must therefore account for *all* segments, either as normal acoustic-phonetic units or as the anti-phone $\bar{\alpha}$. In the example shown in Figure 1, the top path therefore would map feature vectors $\{a_2, a_4\}$ to $\bar{\alpha}$, whereas the bottom path would only map feature $\{a_3\}$ to $\bar{\alpha}$.

We can avoid classifying all the segments in the search space by recognizing that $P(XY|\bar{\alpha})$, the probability that *all* segments are not a lexical unit, is a constant K, and has no effect on decoding. Assuming independence between X and Y, noting that $P(Y|W)$ depends only on $\bar{\alpha}$, we can decompose and rearrange $P(XY|W)$

$$P(XY|W) = P(X|W)P(Y|\bar{\alpha})\frac{P(X|\bar{\alpha})}{P(X|\bar{\alpha})} = K\frac{P(X|W)}{P(X|\bar{\alpha})}$$

Thus, when we consider a particular segmentation S we need only concern ourselves with the N_S feature vectors corresponding to S, but we must combine *two* terms for each segment s_i. The first term is the standard phonetic likelihood $P(x_i|\alpha)$. The second term is the likelihood that the segment is the anti-phone unit, $P(x_i|\bar{\alpha})$. The net

result which must be maximized during search is:

$$W^* = \arg\max_{W,S} \prod_{i=1}^{N_S} \frac{P(x_i|W)}{P(x_i|\tilde{\alpha})} P(s_i|W)P(W)$$

Note that this formulation remains the same whether context-independent or context-dependent modelling is used. The term $P(x_i|W)$ would be reduced accordingly.

4. MODELLING LANDMARKS

In addition to modelling segments, it is often desirable to provide additional information about segment boundaries, or landmarks. If we call the feature-vectors extracted at landmarks Z, we must now consider the joint space XYZ as our observation space. It thus becomes necessary to estimate the probability $P(XYZ|SW)$. If we assume independence between the feature vectors XY representing segments and Z representing landmarks, we can further simplify:

$$P(XYZ|SW) = P(XY|SW)P(Z|SW)$$

If Z corresponds to a set of observations taken at landmarks or boundaries, then a particular segmentation will assign some of the landmarks to *transitions* between lexical units, while the remainder will be considered to occur *internal* to a unit (i.e., within the boundaries of a hypothesized segment). Since any segmentation accounts for *all* of the landmark observations Z, there is no need for the normalization criterion discussed for segment-based feature vectors. If we assume independence between the N_Z individual feature-vectors in Z, $P(Z|SW)$ can be written as

$$P(Z|SW) = \prod_{i=1}^{N_Z} P(z_i|SW)$$

where z_i is the feature vector extracted at the i^{th} landmark. Again, there is no assumption about whether context-independent or context-dependent (diphone) boundary models are used.

5. EXPERIMENTS

Our initial evaluations of this framework were based on phonetic recognition experiments using the TIMIT corpus [3]. Models were built using the TIMIT 61 label set and collapsed down to the 39 labels used by others to report recognition results [4, 7, 8, 14, 15, 21]. Models were trained on the designated training set of 462 speakers, and results are reported on the 24 speaker core test set. A 50 speaker development set (taken from the remaining 144 speakers in the full test set) was used for intermediate experiments so that the core test set was used only for final testing. Reported results are phonetic accuracy which includes substitution, deletion, and insertion errors. The language model used in all experiments was a phone bigram based on the training data with perplexity 15.8 on the development set (using 61 labels). A single parameter (optimized on the development set) controlled the trade-off between insertions and deletions.

All utterances were represented by 14 Mel-scale cepstral coefficients (MFCCs) and log energy, computed at 5 msec intervals. Acoustic landmarks were determined by looking for local maxima in spectral change in the MFCCs [22]. Segment networks were created by fully connecting landmarks within acoustically stable regions. An analysis of the networks showed that on the development set there were 2.4 boundaries per transcription boundary and 7.0 segments per transcription segment on average.

Our research was greatly facilitated by SAPPHIRE, a graphical speech analysis and recognition tool based on Tcl/Tk that is being developed in our group [5]. SAPPHIRE's flexibility and expressiveness allows us to quickly test novel ideas and frameworks.

5.1. Context-Independent Recognition

The first set of experiments we performed used 62 labels (61 TIMIT labels plus the anti-phone "not") to explore context-independent (CI) phonetic recognition using segment-based information only. The feature vector consisted of MFCC and energy averages over segment thirds as well as two derivatives computed at segment boundaries. Duration was also included, as was a count of the number of internal landmarks in the segment. The resulting segment feature vector contained 77 dimensions. Mixtures of up to 50 diagonal Gaussians (400 for the anti-phone) were used to model the phone distributions on the training data. An initial principal components analysis (PCA) was done to normalize the feature space for the mixture generation (which uses K-means clustering as an initial step), though no dimensionality reduction was done. In order to reduce training computation, 20% of the possible anti-phone examples were randomly selected to train the anti-phone model. The CI segment models achieved 64.1% accuracy on the core test set.

5.2. Context-Dependent Recognition

The second set of experiments we performed used a set of context-dependent (CD) diphone models based on feature vectors extracted at hypothesized landmarks. The feature vector consisted of eight averages of MFCC and energy resulting in a 120 dimensional feature vector [14]. PCA was used to normalize the feature space and reduce the dimensionality to 50. A set of 1000 diphone classes (transition and internal) was created based on frequency of occurrence in the training data and simple similarity measures. Up to 50 mixture of diagonal Gaussians were used to model each class. When the diphone models were used by themselves, they achieved a phonetic recognition accuracy of 67.2% on the core test set. When combined with the CI segment models, the accuracy rose to 69.5%.

6. DISCUSSION

As shown in Table 1, there are a number of published results on phonetic recognition using the core test set. There are still differences regarding the complexity of the acoustic and language models, thus making a direct comparison somewhat difficult. Nevertheless, we believe our results are competitive with those obtained by others, and that our performance will improve when we increase the complexity of our models. Internally, both the CI and CD results (64.1 and 69.5%) represent a significant improvement over our previously reported results of 55.3 and 68.5%, respectively [14]. Our previous CD results were achieved by hypothesizing segment boundaries at every frame and performing an exhaustive segment-based search.

Group	Description	% Accuracy
Goldenthal [4]	Trigram, Triphone STM	69.5
Lamel et al. [7]	Bigram, Triphone CDHMM	69.1
Mari et al. [12]	Bigram, 2nd order HMM	68.8
Robinson [15]	Bigram, Recurrent Network	73.4
SUMMIT	Bigram, Diphone	69.5

Table 1: Reported recognition accuracies on the TIMIT core test set.

The word recognition experiments we have performed to date have shown a consistent increase in word accuracy as well. In addition, we have been able to reduce the number of parameters which need to be optimized for recognition. For example, the weights between the segment, boundary, and language model components all optimize to 1.0, whereas in the past, we have optimized each separately.

The framework we have outlined in this paper provides flexibility to explore the relative advantages of segment versus landmark representations. As we have shown, it is possible to use only segment-based feature vectors, or landmark-based feature vectors (which could reduce to frame-based processing), or a combination of both.

The normalization criterion used for segment-based decoding can be interpreted as a likelihood ratio. Acoustic log likelihood scores are effectively normalized by the anti-phone. Phones which score better than the anti-phone will have a positive score, while those which are worse will be negative. In cases of segments which are truly not a phone, the phone scores are typically all negative. Note that the anti-phone is not used during lexical access. Its only role is to serve as a form of normalization for the segment scoring. In this way, it has similarities with techniques being used in word-spotting, which compare acoustic likelihoods with those of "filler" models [17, 18, 20]. The likelihood or odds ratio was also used by Cohen to use HMMs for segmenting speech [1].

The independence assumption between X and Y made to enable efficient decoding is somewhat suspect since overlapping segments are likely correlated with each other. It would therfore be worth examining alternative methods for modelling the joint XY space.

The framework holds whether or not the segmentation is done implicitly or explicitly, or whether the segmentation space is exhaustive, or restricted in some way. The experiments reported here used a constrained network, since this is what we use to achieve near real-time performance for our understanding systems. We are exploring alternative segmentation frameworks to better understand the computation vs. performance tradeoff.

The anti-phone unit we have used in these experiments was based on a single unit which was required to model all possible forms of non-phonetic segments. We have begun to explore the use of multiple anti-phone units to provide better discrimination between "good" and "bad" phones. Finally, we plan to explore CD segment models to improve upon our current performance with diphone models.

7. REFERENCES

1. J. Cohen. Segmenting speech using dynamic programming. *Journal of the Acoustic Society of America*, 69(5):1430–1438, May 1981.
2. V. Digilakis, J. 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.
3. J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallet, and N. Dahlgren. The DARPA TIMIT acoustic-phonetic continuous speech corpus CDROM. NTIS order number PB91-505065, October 1990.
4. W. Goldenthal. Statistical trajectory models for phonetic recognition. Technical report MIT/LCS/TR-642, MIT Lab. for Computer Science, August 1994.
5. L. Hetherington and M. McCandless. SAPPHIRE: An extensible speech analysis and recognition tool based on Tcl/Tk. In *these proceedings*.
6. W. Holmes and M. Russell. Modeling speech variability with segmental HMMs. In *Proc. ICASSP*, pages 447–450, Atlanta, GA, May 1996.
7. L. Lamel and J.L. Gauvain. High performance speaker-independent phone recognition using CDHMM. In *Proc. Eurospeech*, pages 121–124, Berlin, Germany, September 1993.
8. K.F. Lee and H.W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. ASSP*, 37(11):1641–1648, November 1989.
9. H. Leung, I. Hetherington, and V. Zue. Speech recognition using stochastic segment neural networks. In *Proc. ICASSP*, pages 613–616, San Francisco, CA, March 1992.
10. A. Ljolje. High accuracy phone recognition using context clustering and quasi-triphone models. *Computer Speech and Language*, 8(2):129–151, April 1994.
11. J. Marcus. Phonetic recognition in a segment-based HMM. In *Proc. ICASSP*, pages 479–482, Minneapolis, MN, April 1993.
12. J.F. Mari, D. Fohr, and J.C. Junqua. A second-order HMM for high performance word and phoneme-based continuous speech recognition. In *Proc. ICASSP*, pages 435–438, Atlanta, GA, May 1996.
13. M. Ostendorf and S. Roucos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Trans. ASSP*, 37(12):1857–1869, December 1989.
14. M. Phillips and J. Glass. Phonetic transition modelling for continuous speech recognition. *J. Acoust. Soc. Amer.*, 95(5):2877, June 1994.
15. A. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Trans. Neural Networks*, 5(2):298–305, March 1994.
16. T. Robinson, M. Hochberg, and S. Renals. IPA: Improved phone modelling with recurrent neural networks. In *Proc. ICASSP*, pages 37–40, Adelaide, Australia, April 1994.
17. J. Rohlicek, W. Russell, S. Roucos, and H. Gish. Continuous hidden Markov modelling for speaker-independent word spotting. In *Proc. ICASSP*, pages 627–630, Glasgow, Scotland, May 1989.
18. R. Rose and D. Paul. A hidden Markov model based keyword recognition system. In *Proc. ICASSP*, pages 129–132, Albuquerque, NM, April 1990.
19. S. Roucos, M. Ostendorf, H. Gish, and A. Derr. Stochastic segment modelling using the Estimate-Maximize algorithm. In *Proc. ICASSP*, pages 127–130, New York, NY, 1988.
20. J. Wilpon, L. Rabiner, C.H. Lee, and E. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. ASSP*, 38(11):1870–1878, November 1990.
21. S. Young and P. Woodland. State clustering in hidden Markov model-based continuous speech recognition. *Computer Speech and Language*, 8(4):369–383, October 1994.
22. V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Recent progress on the SUMMIT system. In *Proc. Speech and Natural Language Workshop*, pages 380–384, Hidden Valley, PA, June 1990. Morgan Kaufmann.

EXHIBIT C

REAL-TIME TELEPHONE-BASED SPEECH RECOGNITION IN THE JUPITER DOMAIN

James R. Glass, Timothy J. Hazen, and I. Lee Hetherington

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 USA

ABSTRACT

This paper describes our experiences with developing a real-time telephone-based speech recognizer as part of a conversational system in the weather information domain. This system has been used to collect spontaneous speech data which has proven to be extremely valuable for research in a number of different areas. After describing the corpus we have collected, we describe the development of the recognizer vocabulary, pronunciations, language and acoustic models for this system, the new weighted finite-state transducer–based lexical access component, and report on the current performance of the recognizer under several different conditions. We also analyze recognition latency to verify that the system performs in real time.

1. INTRODUCTION

Over the past year and a half, we have developed a telephone-based, weather information system called JUPITER [14], which is available via a toll-free number for users to query a relational database of current weather conditions using natural, conversational speech.[1] Using information obtained from several different internet sites, JUPITER can provide weather forecasts for approximately 500 cities around the world for three to five days in advance, and can answer questions about a wide range of weather properties such temperature, wind speed, humidity, precipitation, sunrise etc., as well as weather advisory information.

The JUPITER system makes use of our GALAXY conversational system architecture which incorporates speech recognition, language understanding, discourse and dialog modelling, and language generation [12]. JUPITER has been particularly useful for our research on displayless interaction, information on demand, and robust spontaneous speech recognition and understanding. Since we attempt to understand all queries (i.e., not spot words), and do not constrain the user at any point in the dialog, it is crucial to have a high accuracy speech recognizer that covers, as much as possible, the full range of user queries. This paper describes our work in developing a robust recognizer in this domain.

When the system was first deployed in late April 1997, the error rates of our recognizer initially more than tripled our laboratory baselines, due in part to the mismatch between the laboratory training and actual testing conditions. The real data had a much larger variation in environment and channel conditions (often with very poor signal conditions), as well as a much wider range of speakers (we had no children in our training data for example, and had mainly trained on native speakers without regional accents), speaking style (spontaneous speech vs. read speech), language (both for within-domain queries, and out-of-domain queries), and other artifacts such as non-speech sounds and clipped speech due to the user interface (we do not currently allow for barge-in).

As we have collected more data we have been able to better match the users' vocabulary, and build more robust acoustic and language models. The result is that we have steadily reduced word and sentence error rates, to the point of cutting the initial error rates by over two thirds. In this paper, we describe the methods we have used to develop this recognizer and report on the lessons we have learned in moving from a laboratory environment to dealing with real data collected from real users. Our experience has shown us clearly that while there is no data like more data, there is also no data like *real* data!

2. CORPUS

Several different methods have been employed to gather data for the JUPITER weather information system. Beginning in February and March 1997, we created an initial corpus of approximately 3,500 read utterances collected from a variety of local telephone handsets and recording environments, augmented with over 1,000 utterances collected in a wizard environment [14]. These data were used to create an initial version of a conversational system which users could call via a toll-free number and ask for weather information. The benefit of this setup is that it provides us with a continuous source of data from users interested in obtaining information. Currently, we average over 70 calls per day, and have recorded and orthographically transcribed over 60,000 utterances from over 11,000 callers, all without widely advertising the availability of the system. On average, each call contains 5.6 utterances, and each utterance has an average of 5.2 words. The data are continually orthographically transcribed (seeded with the system hypothesis), and marked for obvious non-speech sounds, spontaneous speech artifacts, and speaker type (male, female, child) [4].

3. VOCABULARY

The vocabulary used by the JUPITER system has evolved as periodic analyses are made of the growing corpus. The current vocabulary contains 1893 words, including 638 cities and 166 countries. Nearly half of the vocabulary contains geography-related words.

This research was supported by DARPA under contract N66001-96-C-8526, monitored through Naval Command, Control and Ocean Surveillance Center.

[1] In the United States and Canada please call 888 573-8255 or visit http://www.sls.lcs.mit.edu/jupiter.

0-7803-5041-3/99 $10.00 © 1999 IEEE

can_you	when_is	never_mind
do_you	what_about	clear_up
excuse_me	what_are	heat_wave
give_me	what_will	pollen_count
going_to	how_about	warm_up
you_are	i_would	wind_chill

Table 1: Examples of multi-word units in the JUPITER domain.

The design of the geography vocabulary was based on the cities for which we were able to provide weather information, as well as commonly asked cities. Other words were incorporated based on frequency of usage and whether or not the word could be used in a query which the natural language component could understand. The 1893 words had an out-of-vocabulary (OOV) rate of 2.0% on a 2506 utterance test set.

Since the recognizer makes use of a bigram grammar in the forward Viterbi pass, several multi-word units were incorporated into the vocabulary to provide for greater long-distance constraint and, in some cases, to allow for specific pronunciation modelling. This would allow for explicit modelling of word sequences such as "going to" or "give me" to be pronounced as "gonna" or "gimme" respectively. Common contractions such as "what's" were represented as multi-word units (e.g., "what_is") to reduce language model complexity, and because these words were often a source of transcription error anyway. Additional multi-word candidates were identified using a mutual information criterion which looked for word sequences which were likely to occur together. Table 1 shows examples of multi-word units in the current vocabulary.

4. PHONOLOGICAL MODELING

In the current JUPITER recognizer, words are initially represented as sequences of phonetic units augmented with stress and syllabification information. The initial baseform pronunciations are drawn from the LDC PRONLEX dictionary. The baseforms are represented using 41 different phonetic units with three possible levels of stress for each vowel. The baseforms have also been automatically syllabified using a basic set of syllabification rules. After drawing the pronunciations for the JUPITER vocabulary from the PRONLEX dictionary, all baseform pronunciations were then verified by hand. Vocabulary words missing from the dictionary were hand coded. Alternate pronunciations are explicitly provided for some words. In addition to the standard pronunciations for single words provided by PRONLEX, the baseform file was also augmented with common multi-word sequences which are often reduced, such as "gonna", "wanna", etc.

A series of phonological rules were applied to the phonetic baseforms to expand each word into a graph of alternate pronunciations. These rules account for many different phonological phenomena such as place assimilation, gemination, epenthetic silence insertion, alveolar stop flapping, and schwa deletion. These phonological rules utilize stress, syllabification, and phonetic context information when proposing alternate pronunciations. We have made extensive modification to these rules, based on our examination of the JUPITER data.

The final pronunciation network does not represent the words using the original 41 phonetic units utilized in PRONLEX. Instead, a set of 105 different units were used which include sub-phonetic, supra-phonetic and non-phonetic units in addition to standard phonetic units. For example, the recognizer treats most within-syllable vowel-semivowel sequences and some semivowel-vowel sequences as single units in order to better model the highly correlated dynamic characteristics of these sequences. Thus, the phonetic sequence [ow] followed by [r] is represented as a single segmental unit [or]. The recognizer also incorporates various non-phonetic units to account for non-linguistic speech transitions and speech artifacts, silences, and non-speech noise. The 105 units also retain two levels of stress for each vowel unit. An example pronunciation graph for the word "reports" is shown in Figure 1.

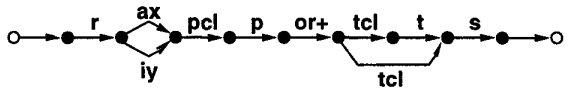

Figure 1: Pronunciation graph for the word "reports."

The arcs in the pronunciation graph can further be augmented with transition weights which give preference to more likely pronunciations and penalize less likely pronunciations. For JUPITER these weights were set using an error correcting algorithm on development data [13]. This algorithm adjusted the arc weights in an iterative fashion in order to reduce the error rate of the recognizer on development data.

5. LANGUAGE MODELLING

A class bigram language model was used in the forward Viterbi search, while a class trigram model was used in the backwards A^* search to produce the 10-best outputs for the natural language component. A set of nearly 200 classes were used to improve the robustness of the bigram. The majority of the classes involved grouping cities by state or country (foreign), in order to encourage agreement between city and state. In cases where a city occurred in multiple states or countries, separate entries were added to the lexicon (e.g., Springfield, Illinois vs. Springfield, Massachusetts). Artificial sentences were created in order to provide complete coverage of all of the cities in the vocabulary. Other classes were created semi-automatically using a relative entropy metric to find words which shared similar conditional probability profiles.

Since filled pauses (e.g., uh, um) occurred both frequently and predictably (e.g., start of sentence), they were incorporated explicitly into the vocabulary, and modelled by the bigram and trigram. Original orthographies were modified for training and testing purposes by removing non-speech and clipped word markers. When trained on a 26,000 utterance set, and tested on a 2506 utterance set the word-class bigram and trigram had perplexities of 18.4 and 17.1, respectively. These are slightly lower than the respective *word* bigram and trigram perplexities of 19.5 and 18.8. Note that the class bigram also improved the speed of the recognizer as it had 22% fewer connections to consider during the search.

6. ACOUSTIC MODELLING

The JUPITER system makes use of the segment-based SUMMIT recognizer which can utilize acoustic models based on segments or landmarks [3]. The nature of the acoustic models has varied over the course of system development, depending in large part on

the amount of available training data. The current JUPITER configuration makes use of context-dependent landmark-based diphone models which require the training of both *transition* and *internal* diphone models. Internal diphones model the characteristics of landmarks occurring within the boundaries of a hypothesized phonetic segment, while transition diphones model the characteristics of landmarks occurring at the boundary of two hypothesized phonetic segments.

Given the 105 phonetic units used in the JUPITER system, and the constraints of the full pronunciation graph, there were 4,822 possible diphone transition models and 105 internal models needed. We have explored two different methods of modelling transitions. The first method trained models for frequently occurring transitions, and used one "catch-all" model for remaining transitions. This method worked well, and was simple to train. We currently use a reduced set of 782 equivalence classes which were determined manually to insure that an adequate amount of training existed for each class and that the elements of each class exhibited contextual similarity. This method performs slightly better than the "catch-all" method.

For each landmark, 14 MFCC averages were computed for 8 different regions surrounding the landmark, creating 112 different features. This initial feature set was then reduced from 112 features to 50 features using principal component analysis. The acoustic models for each class modeled the 50 dimensional feature vectors using diagonal Gaussian mixture models. Each mixture model consisted of a variable number of mixture components, dependent on the number of available training vectors for that class, with a maximum of 50 mixture components.

The diphone models were trained on a subset of data which excludes utterances with out-of-vocabulary words, clipped speech, cross-talk, and various types of noise. The training data also excludes all speech from speakers deemed to have a strong foreign accent. The full set of within-domain training utterances used for acoustic modelling consisted of 20,064 utterances, which was 76% of the available data at the time.

7. LEXICAL ACCESS

We have recently re-implemented the lexical access search components of SUMMIT to use weighted finite-state transducers with the goals of increasing recognition speed while allowing more flexibility in the types of constraints. We view recognition as finding the best path(s) through the composition $A \circ U$, where A represents the scored (on demand) acoustic segment graph and U the complete model of an utterance from acoustic model labels through the language model. We compute $U = C \circ P \circ L \circ G$, where C maps context-independent labels on its right to context-dependent (diphone in the case of JUPITER) labels on its left, P applies phonological rules, L is the lexicon mapping pronunciations to words, and G is the language model. Any of these transductions can be weighted. A big advantage of this formulation is that the search components operate on a *single* transducer U; the details of its composition are not a concern to the search. As such, U can be precomputed and optimized in various ways or it can be computed on demand as needed. This use of a cascade of weighted finite-state transducers is heavily inspired by work at AT&T [8, 10].

We have achieved our best recognition speed by precomputing $U = C \circ \text{minimize}(\text{determinize}((P \circ L) \circ G))$ for G a word-class bigram. This yields a deterministic (modulo homophones), minimal transducer that incorporates all contextual, phonological, lexical, and language model constraints [8]. For JUPITER, U has 89,452 states and 699,172 arcs. We apply a word-class trigram and compute N-best in a second pass utilizing an A^* search.

For greater system flexibility, we can compute $U = (C \circ \text{minimize}(\text{determinize}(P \circ L))) \circ G$, performing the composition with G on the fly during the search. For example, the use of a dynamic language model that changes during a dialogue would require this approach. However, with on-the-fly composition we have found that the system runs about 40% slower than for the fully composed and optimized U.

Test Set	# Utts	WER	SER
Standard	2506	24.4	43.0
In domain	1806	13.1	28.6
Male (In domain)	1290	9.8	24.1
Female (In domain)	274	13.6	31.8
Child (In domain)	242	26.3	48.8
Out of domain	700	61.8	80.1
Non-native (In domain)	3225	29.9	60.1
Expert (In domain)	354	2.3	7.1

Table 2: Current JUPITER performance on various test sets.

8. EXPERIMENTS

Over the course of the past year the JUPITER recognizer has had a steady improvement in its performance; this has been a result of both an increase in training data and improvements to the system's modeling techniques. The test data consists of sets of calls randomly selected over our data collection period. The current test set consists of 2506 utterances, of which 1806 were considered to be "in domain" as they were covered by the vocabulary, were free of partial words, crosstalk, etc. Of these sentences, 1290 were from male speakers, 274 from females, and 242 from children. Table 2 shows the performance of the JUPITER recognizer on this test set using word error rate (WER) and sentence error rate (SER) as the evaluation metrics.[2] As can be seen in the table, the system tends to perform reasonably when it encounters queries spoken by adults without a strong accent, that are covered by the domain, and that do not contain spontaneous, or non-speech artifacts. Females had 50% more word errors than males, while children had 300% more word errors than males. This is probably a reflection of the lack of training material for females and children. The system has considerable trouble (64.5% WER) with "out of domain" utterances containing out-of-vocabulary words, partial words, crosstalk, or other disrupting effects. This rate is artificially high, however, due to the nature of the alignment procedure with reference orthographies (e.g., partial words *always* cause an error for example, due to the nature of our mark-up scheme).

Table 2 also shows the performance on speakers judged to have strong foreign accents, who were not included in the standard test set. These data consisted of 3,225 in-domain utterances, and had an error rate more than double the baseline in-domain error rate. Finally, we also evaluated the recognizer on "expert" users (i.e., mainly staff in our group) who have considerable experience using the JUPITER system, but were not used for training. The system had extremely small error rates for these users. This behavior

[2]These error rates are slightly different from those reported in [4]. The reason is that we have increased pruning to achieve real-time performance.

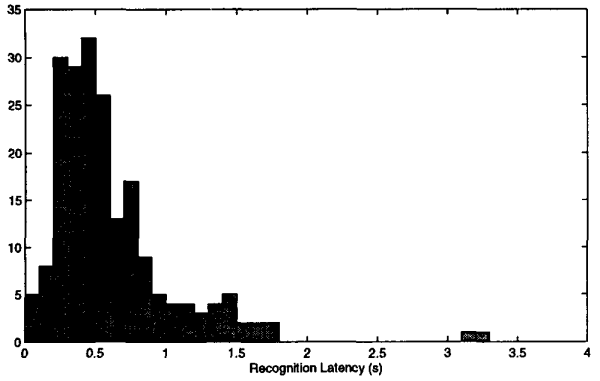

Figure 2: Histogram of JUPITER recognition latency.

is typical of users who become familiar with the system capabilities (a case of users adapting to the computer!).

Since JUPITER is a conversational system, rapid system response is critical. We consider a recognizer to run in real time if its latency (time after utterance is complete) is independent of utterance duration. An initial analysis of latency showed that while the latency was generally less than 1s, the worst cases took substantially longer. Not surprisingly, most of the worst cases were due to out-of-domain utterances containing out-of-vocabulary words. To combat the worst-case latency, we have added count-based beam pruning to limit the number of active nodes kept at any given point in time. Previously, we limited the beam solely with a score-based pruning threshold. With aggressive count-based pruning on a 300MHz Pentium II, we find a correlation coefficient between latency and utterance length of only -0.08, meaning that they are independent and we are achieving real-time performance. Figure 2 shows a histogram of the latency: 85% of the time the latency is less than 1s, and 99% of the time it is less than 2s.

9. DISCUSSION & FUTURE WORK

The speech recognizer described in this paper is only one component of the full JUPITER conversational system [11, 14]. The current interface between the recognizer and our language understanding component is via an N-best interface. Although we have reported only first-choice error rates in this paper, the understanding error rates are typically better, since many word confusions do not impact understanding.

There remain a considerable number of ongoing areas of research we are presently pursuing, which should help improve performance. Recent developments in probabilistic segmentation [7], near-miss modelling [1], heterogeneous classifiers [5], and tighter integration of linguistic knowledge [2], have shown improvements in our JUPITER baseline, although they have not yet been propagated to the data collection system.

The system to date has used a pooled speaker model for all acoustic modelling. It should be possible to achieve gains through speaker normalization, short-term speaker adaptation, and better adaptation to the channel conditions of individual phone calls. Adaptation may also be useful to help improve performance on non-native speakers. Since a phone call could have multiple speakers, we are exploring within-utterance consistency techniques that have given us gains elsewhere [6].

The data collection efforts have produced a gold-mine of spontaneous speech effects which are often a source of both recognition and understanding errors. For example, partial words typically cause problems for the speech recognizer. Another source of recognition errors is out-of-vocabulary words, which are often cities not covered in the vocabulary. These issues have caused us to begin work in confidence scoring, which was an area we had not previously addressed [9]. Finally, we plan to explore the use of dynamic vocabulary and language models, which may help to alleviate some of the unknown city problems.

10. ACKNOWLEDGMENTS

The JUPITER data collection effort was coordinated by Joe Polifroni, and heroically transcribed by Sally Lee. Jon Yi syllabified the LDC PRONLEX dictionary.

11. REFERENCES

[1] J. Chang, *Near-Miss Modeling: A Segment-Based Approach to Speech Recognition*. Ph.D. thesis, MIT, May 1998.

[2] G. Chung and S. Seneff, "Improvements in speech understanding accuracy through the integration of hierarchical linguistic, prosodic, and phonological constraints in the JUPITER domain," in *Proc. ICSLP*, Sydney, 1998.

[3] J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. ICSLP*, Philadelphia, PA, pp. 2277–2280, 1996.

[4] J. Glass and T. Hazen, "Telephone-based conversational speech recognition in the Jupiter domain," in *Proc. ICSLP*, Sydney, 1998.

[5] A. Halberstadt and J. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," in *Proc. ICSLP*, Sydney, 1998.

[6] T. J. Hazen, *The Use of Speaker Correlation Information for Automatic Speech Recognition*. Ph.D. thesis, MIT, January 1998.

[7] S. Lee and J. Glass, "Real-time probabilistic segmentation for segment-based speech recognition," in *Proc. ICSLP*, Sydney, Australia, 1998.

[8] M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. Pereira, "Full expansion of context-dependent networks in large vocabulary speech recognition," in *Proc. ICASSP*, Seattle, WA, vol. 2, pp. 665–668, May 1998.

[9] C. Pao, P. Schmid, and J. Glass, "Confidence scoring for speech understanding," in *Proc. ICSLP*, Sydney, Australia, 1998.

[10] F. Pereira and M. Riley, "Speech recognition by composition of weighted finite automata," in *Finite-State Language Processing* (E. Roche and Y. Schabes, eds.), pp. 431–453, Cambridge, MA: MIT Press, 1997.

[11] J. Polifroni, S. Seneff, J. Glass, and C. Pao, "Evaluation methodology for a telephone-based conversational system," in *Proc. First Int'l. Conference on Language Resources and Evaluation*, Granada, Spain, pp. 43–49, 1998.

[12] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "Galaxy-II: A reference architecture for conversational system development," in *Proc. ICSLP*, Sydney, Australia, 1998.

[13] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff, "The SUMMIT speech recognition system: Phonological modelling and lexical access," in *Proc. ICASSP*, Albuquerque, NM, pp. 49–52, 1990.

[14] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid, "From interface to content: Translingual access and delivery of on-line information," in *Proc. Eurospeech*, Rhodes, Greece, pp. 2047–2050, 1997.

EXHIBIT D

OPTIMAL SUBSET SELECTION FROM TEXT DATABASES

Jilei Tian, Jani Nurminen and Imre Kiss

Multimedia Technologies Laboratory
Nokia Research Center, Tampere, Finland
{jilei.tian, jani.k.nurminen, imre.kiss}@nokia.com

ABSTRACT

Speech and language processing techniques, such as automatic speech recognition (ASR), text-to-speech (TTS) synthesis, language understanding and translation, will play a key role in tomorrow's user interfaces. Many of these techniques employ models that must be trained using text data. In this paper, we introduce a novel method for training set selection from text databases. The quality of the training subset is ensured using an objective function that effectively describes the coverage achieved with the strings in the subset. The validity of the subset selection technique is verified in an automatic syllabification task. The results clearly indicate that the proposed systematic selection approach maximizes the quality of the training set, which in turn improves the quality of the trained model. The presented idea can be used in a wide variety of language processing applications that require training with text databases.

1 INTRODUCTION

Most automatic speech recognition (ASR) and text-to-speech (TTS) systems contain models that have to be trained with text data. Typical examples can be found from many parts of the systems. In pronunciation modeling, some data-driven approach, such as neural network based methods or decision tree based methods [6], are often applied, especially for languages like English. These statistical models are trained using a pronunciation dictionary containing grapheme-to-phoneme entries. In text-based language identification [8], the model is trained using a multilingual text corpus that consists of word entries from the target languages. In the data-driven syllabification task [7], the model is trained using text-based pronunciations and the corresponding syllable structures.

In all data-driven approaches, the selection of a suitable training set can be regarded as a very important step in the training process. In general, the performance of any trained model depends quite strongly on the quality of the text data used in the training. With text-based data, the importance of the training set selection is very pronounced since the generation of the training data entries is often very time and resource consuming and requires language-specific skills. In this paper, we show that systematic training set selection results in enhanced model performance and/or offers the possibility to use a smaller training set size. In practice, the reduced training set size brings two significant additional benefits. First, the amount of manual annotation work is reduced, which in turn decreases the probability of errors and inconsistencies in the annotations. Second, the memory consumption and the computational load caused by the training process are lowered. In some cases this advantage propagates to the trained model as well; the size of a decision tree model, for example, depends on the size of the training set.

Despite the evident importance of the training set selection, this step is often neglected in practice. Usually, the training set is obtained by collecting a set of random entries from a larger text database or by decimating a sorted corpus. The drawback of these solutions is that the amount of meaningful information in the selected text data set is not maximized. The random selection method is rather coarse and does not produce consistent results. The method of decimating a sorted data corpus, on the other hand, only uses a limited number of the initial characters of the strings and thus does not guarantee good performance.

In this paper, we present a method that can quasi-optimally select a subset from a text database in such a manner that the text coverage is maximized. To achieve this, we define an objective function that is optimized in the subset selection. The objective function measures the "subset distance" using the generalized Levenshtein distances between the text strings. This paper also introduces an algorithm for optimizing the objective function. For practical applications with large databases, the algorithm can be modified in order to speed up the processing or to lower the memory consumption, but the main idea and the objective function will remain useful in all cases. To demonstrate the usefulness of the proposed approach, we evaluate it in the syllabification task.

The text subset selection method introduced in this paper can be used in a wide variety of different applications. One good example is the language identification task [8], in which the proposed approach makes it possible to easily balance the number of training set entries from each target language while at the same time giving a good coverage for every target language. In addition to the training set selection task discussed extensively in this paper, it is possible to employ the same techniques for clustering a text database. Moreover, when used together with a meaningful distance measure, such as the generalized Levenshtein distance, the proposed approach enables the use of vector quantization techniques on text data.

The remainder of the paper is organized as follows. We first describe the generalized Levenshtein distance and introduce the basic principles of the text database selection algorithm in Section 2. In Section 3, we describe the syllabification task used as the practical example by briefly reviewing the syllable structure grammar and the neural network based syllabification method. The performance of the proposed subset selection approach is evaluated in the

syllabification task in Section 4. Finally, some concluding remarks are presented in Section 5.

2 SELECTION ALGORITHM

In order to be able to select a subset from a text database in a systematic and meaningful manner, an objective function measuring the quality of the subset must be defined. The objective function should somehow measure the similarity or the dissimilarity of the entries. In the proposed approach, we base the objective function on the generalized Levenshtein distance. In this section, we first describe the basic properties of this distance measure and then continue by defining an objective function measuring the average distance within a subset and by introducing an algorithm for selecting subsets of different sizes in a quasi-optimal manner.

2.1 Generalized Levenshtein distance

The generalized Levenshtein distance (GLD) is defined as the minimum cost of transforming one string into another by means of a sequence of basic transformations: insertion, deletion and substitution [4]. The transformation cost is determined by the costs assigned to each basic transformation.

Let x and y be strings of length m and n, respectively, whose symbols belong to a finite alphabet of size s. Let x_i be the ith symbol of string x, with $1 \le i \le m$, and $x(i)$ be the prefix of the string x of length i, i.e. the substring containing the first i symbols of x. In addition, let $d(i,j)$ be the distance between $x(i)$ and $y(j)$, and ε be an empty string. Furthermore, we denote by $w(a,b)$, $w(a,\varepsilon)$ and $w(\varepsilon,b)$ the cost of substituting the symbol a with the symbol b, the cost of deleting a and the cost of inserting b, respectively. The distance $d(m,n)$ is recursively computed based on the definitions of $d(0,0)$, $d(i,0)$ and $d(0,j)$ ($i = 1\ldots m$, $j = 1\ldots n$), representing the initial distance, the cost of deleting the prefix $x(i)$ and the cost of inserting the prefix $y(j)$, respectively, as follows:

$$d(0,0) = 0$$
$$d(i,0) = d(i-1,0) + w(x_i, \varepsilon) \qquad \forall i = 1\ldots,m \tag{1}$$
$$d(0,j) = d(0,j-1) + w(\varepsilon, y_j) \qquad \forall j = 1\ldots,n$$

$$d(i,j) = \min\begin{cases} d(i-1,j) + w(x_i, \varepsilon) \\ d(i,j-1) + w(\varepsilon, y_j) \\ d(i-1,j-1) + w(x_i, y_j) \end{cases} \tag{2}$$

The original Levenshtein distance is characterized by the following costs: $w(a, \varepsilon) = 1$, $w(\varepsilon,b) = 1$, and $w(a, b)$ is 0 if a is equal to b and 1 otherwise. Its generalized version assumes that different costs can be associated to transformations involving different symbols. In the case of an alphabet of s symbols, this requires a table of size (s+1) times (s+1), called the cost table, to store all the substitution, insertion and deletion costs. It can be shown that the defined distance is a metric if the cost table is symmetric.

2.2 Objective function and selection algorithm

In our approach, we measure the quality of a text subset using an objective function based on the generalized Levenshtein distance. As described in Section 2.1, the Levenshtein distance can be used for measuring the distance between any pair of entries. Similarly, the distance for the whole text data set can be calculated by averaging the distances of all the string pairs in the set. Suppose that there are m entries in the database and the ith entry is denoted by $e(i)$. With these definitions, we can compute the overall "subset distance" D as:

$$D = \frac{2 \cdot \sum_{i=1}^{m} \sum_{j>i}^{m} ld(e(i), e(j))}{m \cdot (m-1)}, \tag{3}$$

where $ld(e(i), e(j))$ is the GLD between the ith and jth entries.

Based on the above objective function, it is possible to design an algorithm that selects a subset from a text database in such a manner that the distance D is maximized. The following algorithm recursively constructs the subset by always selecting the new entry that maximizes the distance to the other selected entries.

1. Calculate the Levenshtein distances for all the pairs; $ld(e(i), e(j))$;
2. Initially select the pair that has the largest distance among all pairs in the database,
$$((subset\_e(1), subset\_e(2)) = \underset{(1 \le i \le m, j > i)}{\operatorname{argmax}} \{ld(e(i), e(j))\}. \tag{4}$$
3. Assuming that the selected subset has k entries (in the first time k = 2), the target now is to find the k+1-th entry to the subset. The selection that approximately maximizes the amount of new information brought into the subset can be done using the following formula.
$$p = \underset{(1 \le i \le m)}{\operatorname{argmax}} \left\{ \sum_{j=1, e(i) \ne subset\_e(j)}^{k} ld(e(i), subset\_e(j)) \right\}. \tag{5}$$
The selected entry p is added into the subset as $subset\_e(k+1)$.
4. Repeat step 3 until the preset subset size is reached.

3 EXAMPLE APPLICATION: SYLLABIFICATION TASK

The development of speech synthesizers and speech recognizers often requires working with sub-word units such as syllables [5]. We have earlier described a neural network based approach for the automatic assignment of syllable boundaries in [7]. In this paper, we revisit the topic and use this syllabification task for verifying the usefulness of the proposed subset selection approach. The first part of this section gives some basic information on the task and the second part discusses the neural network approach. The practical results achieved in this task are presented in Section 4.

3.1 Syllable structure

A syllable is a basic unit of word studied on both the phonetic and phonological levels of analysis [2]. The syllable information can be described using grammars [3]. The simplest grammar is the phoneme grammar, where a syllable is tagged with the corresponding phoneme sequence. The consonant-vowel grammar describes a syllable as a consonant-vowel-consonant (CVC) sequence. The syllable structure grammar, on the other hand, divides a syllable into onset, nucleus and coda (ONC) as shown in Figure 1. The nucleus is an obligatory part that can be either a vowel or a diphthong. The onset is the first part of a syllable consisting of consonants and ending at the nucleus of the syllable, e.g. in the syllable [*t eh k s t*], /*t*/ is the onset and the vowel part /*eh*/ is the nucleus. The part of a syllable that follows the nucleus forms the coda. The coda is constructed of consonants, e.g. /*k s t*/ in

our example syllable. The nucleus and coda are combined to form the rhyme of a syllable. A syllable has a rhyme, even if it doesn't have a coda.

In the syllable structure grammar, the consonants are assigned as onset or coda. The ONC representation used in the syllable structure grammar contains more information than the CVC structure for multi-syllable words. The syllable structure grammar was used in [7] and it is also used in this paper.

In the automatic syllabification task, the phoneme sequences are mapped into their ONC representations. The data-driven syllabification model is trained on the mapping information. In the decoding phase, given a phoneme sequence, the ONC sequence is first generated, and then the syllable boundaries are uniquely decided on the ONC sequence. For invalid ONC sequences, a self-correction algorithm [7] can be applied to solve the problem by utilizing certain common linguistic rules. The whole syllabification task can be summarized as follows:

1. Each pronunciation phoneme string in the training set is mapped into the corresponding ONC string, for example:
(word) *text* -> (pronunciation) *t eh k s t* -> (ONC) *O N C C C*
2. The model is trained on the data in the format of "pronunciation -> ONC"
3. Given a pronunciation string, the corresponding ONC sequence is generated using the model. Then, the syllable boundaries are placed at the location starting with symbol "*O*", or with "*N*" if it is not preceded with symbol "*O*".

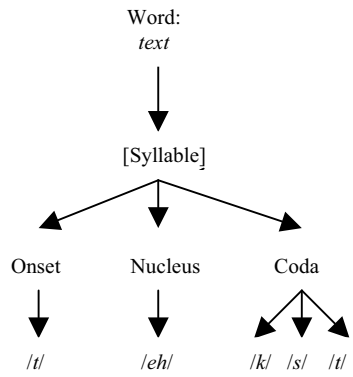

Figure 1. Diagram of the syllable structure grammar.

3.2 Neural network based syllabification approach

The basic neural network based ONC model presented in [7] is a standard multi-layer perceptron (MLP) shown in Figure 2. The input phonemes are presented to the MLP network in a sequential manner. The network gives estimates of ONC posterior probabilities for each presented phoneme. In order to take the phoneme context into account, a number of phonemes on each side of the phoneme in question are also used as inputs to the network. Thus, a window of phonemes is presented to the neural network as input. Figure 2 shows a typical MLP with a context size of w phonemes, $ph_{i-w}...ph_{i+w}$ centered at phoneme ph_i. The centermost phoneme ph_i is the phoneme that corresponds to the output of the network. Therefore, the output of the MLP is the estimated ONC probability $P(onc_k|ph_{i-w},...,ph_{i+w})$ ($onc_k \in \{O,N,C\}$) for the centermost phoneme ph_i in the given context $p_{i-w}...p_{i+w}$. A phonemic null is defined in the phoneme set and is used for representing phonemes to the left of the first phoneme and to the right of the last phoneme in a pronunciation.

The ONC neural network is a fully connected MLP, which uses a hyperbolic tangent sigmoid shaped function in the hidden layer and a softmax normalization function in the output layer. The softmax normalization ensures that the network outputs are in the range [0,1] and sum up to unity,

$$P_i = \frac{e^{y_i}}{\sum_{j=1}^{3} e^{y_j}}. \quad (6)$$

In Equation (6), y_i and P_i denote the ith output value before and after softmax normalization. It has been shown in [1] that a neural network with softmax normalization will approximate class posterior probabilities when trained for one-out-of-N classification and when the network is sufficiently complex and trained to a global minimum. Since the neural network input units are text-valued, the phonemes in the input window need to be transformed to some numeric quantity. This can be done, for example, using an orthogonal codebook representing the alphabet used for the ONC mapping task, as shown in Table 1. The last row in the table is the code for the phonemic null. An important property of the orthogonal coding scheme is that it does not introduce any correlation between the different letters.

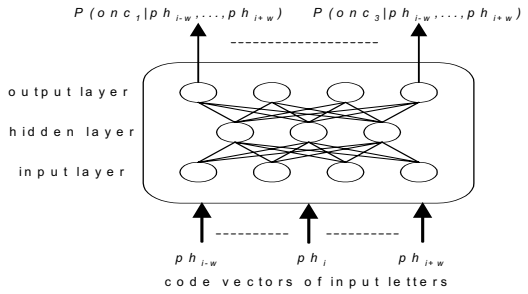

Figure 2. Two-layer neural network architecture.

The ONC neural network is trained using the standard back-propagation (BP) algorithm augmented by a momentum term. Each phoneme with context and the corresponding ONC tag of the pronunciation make up one training example. Weights are updated in a stochastic on-line fashion. All parameters are rounded off to eight bits as this was found sufficient for representing model parameters.

Table 1. Orthogonal phoneme coding scheme.

Letter	Code
aa	100...0000
ae	010...0000
...	...
B	000...1000
P	000...0100
T	000...0010
#	000...0001

The outputs of the ONC neural network approximate the ONC posterior probabilities corresponding to the centermost phoneme. The ONC sequence of a pronunciation is obtained by combining the network outputs for each individual phoneme in the pronunciation. Given a pronunciation with its

phonemic representation, the ONC tag of phoneme ph_i is given by

$$onc = \underset{onc_k}{\operatorname{argmax}}\{P(onc_k \mid ph_{i-w},..., ph_{i+w})\}, \quad (7)$$

where $P(onc_k \mid ph_{i-w},..., ph_{i+w})$ is the network output corresponding to onc_k given the input phonemes $ph_{i-w}...ph_{i+w}$, and variable w denotes the phoneme window context size, respectively. The variable onc takes its values from the set [*O N C*].

4 EXPERIMENTAL RESULTS

The neural network based syllabification method is evaluated using the CMU dictionary for US English. The dictionary contains 10,801 words with their pronunciations and labels with ONC information. The pronunciations and the mapped ONC sequences are extracted to form the training data. The training set is selected from the whole database by using the following methods:

- Decimation of the sorted dictionary (denoted as DECIMATE);
- Subset selection from the text database using the selection approach proposed in this paper (denoted as SELECT).

With both methods, the data not selected to the training set constitutes the test set.

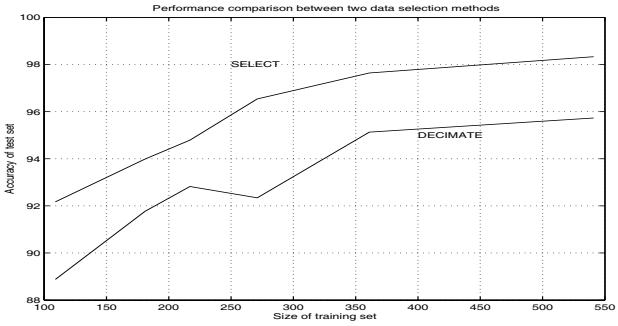

Figure 3. ONC accuracy on test set with different training set sizes using the two data selection methods.

Figure 3 shows the experimental results achieved using the two data selection methods. The efficiency of the training set selection approach can be studied by evaluating the generalization capability. The general rule of thumb is that the more training data is available, the better performance can be expected. However, the selection of the training data affects the generalization capability: if the training data is well selected, the performance can be improved without increasing the size of the training set. The results clearly show that the proposed subset selection technique outperforms the commonly used decimation method; the average improvement achieved using the proposed approach is 38.8%.

Figure 4 illustrates the "subset distance" (see Section 2.2) of datasets extracted using the two different data selection methods: the decimation technique and the proposed selection algorithm. It is easy to see that the average distance *D* is more or less even when the decimation method is used. With the proposed method, the average distance decreases monotonically with increasing data size. Furthermore, the difference between the two methods is large with small subset sizes, and converges to zero when the whole data set is used. Thus, these results indicate that the proposed method can extract data more efficiently, i.e. the selected data has better coverage. Naturally, this explains the better generalization capability of the trained model.

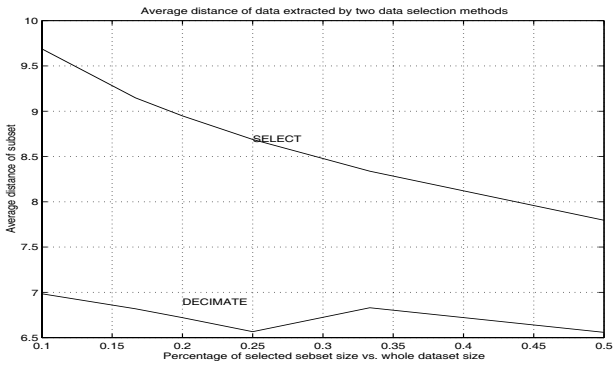

Figure 4. Average distance *D* inside the subsets extracted using the two different data selection methods, with respect to the percentage of the subset size vs. the whole data size.

5 CONCLUSIONS

Training data selection from a text database is a crucial, but often neglected, step in the development of ASR and TTS systems. In this paper, we define an objective function that effectively measures the quality of a selected subset. Moreover, we introduce a subset selection algorithm that optimizes the objective function. Our experimental results obtained in the syllabification task show that the proposed approach is a very promising technique that makes it possible to select subsets with good coverage in a systematic and meaningful way. The presented idea can be used in many different applications that require training with a text database.

6 REFERENCES

[1] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[2] D. Kahn, *Syllable-Based Generalizations in English Phonology*, Doctoral Dissertation, Massachusetts Institute of Technology, USA, 1976.

[3] K. Müller, "Automatic Detection of Syllable Boundaries Combining the Advantages of Treebank and Bracketed Corpora Training", in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, 2001.

[4] E. Ristad and P. Yianilos, "Learning String Edit Distance", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.20, pp.522-532, May, 1998.

[5] R. Sproat, *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Kluwer, Dordrecht, 1998.

[6] J. Suontausta, and J. Häkkinen, "Decision Tree Based Text-to-Phoneme Mapping for Speech Recognition," In *Proceedings of 6th ICSLP*, Beijing, China, 2000.

[7] J. Tian, "Data-Driven Approaches for Automatic Detection of Syllable Boundaries", in *Proceedings of 8th ICSLP*, Jeju Islands, Korea, 2004.

[8] J. Tian, J. Häkkinen, S. Riis, and K. Jensen, "On Text-Based Language Identification for Multilingual Speech Recognition Systems, In *Proceedings of 7th ICSLP*, Denver, USA, 2002.

EXHIBIT E

INTEGRATING SYLLABLE BOUNDARY INFORMATION INTO SPEECH RECOGNITION

Su-Lin Wu, Michael L. Shire, Steven Greenberg, Nelson Morgan

International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA 94704-1198, USA
University of California at Berkeley, Berkeley, CA 94720, USA
{sulin, shire, steveng, morgan}@icsi.berkeley.edu

ABSTRACT

In this paper we examine the proposition that knowledge of the timing of syllabic onsets may be useful in improving the performance of speech recognition systems. A method of estimating the location of syllable onsets derived from the analysis of energy trajectories in critical band channels has been developed, and a syllable-based decoder has been designed and implemented that incorporates this onset information into the speech recognition process. For a small, continuous speech recognition task the addition of artificial syllabic onset information (derived from advance knowledge of the word transcriptions) lowers the word error rate by 38%. Incorporating acoustically-derived syllabic onset information reduces the word error rate by 10% on the same task. The latter experiment has highlighted representational issues on coordinating acoustic and lexical syllabifications, a topic we are beginning to explore.

1. INTRODUCTION

Automatic speech recognition (ASR) systems typically rely upon phoneme- or sub-phoneme-based Hidden Markov models (HMMs) that are concatenated into word and sentence elements. Although syllable-based recognition has been successfully used in several languages (including Spanish [1] and Chinese [2]), the syllable has been not been fully exploited for the automatic recognition of English. In this paper we investigate the possibility that syllabic onsets can be derived from the acoustic speech signal, and that this onset information can be incorporated into the decoding process in a manner sufficient to improve recognition performance.

Evidence from both psychoacoustic and psycholinguistical research [3, 4, 5], as well as a model by one of the authors [6], suggests that the syllable is a basic perceptual unit for speech processing in humans. The syllable was proposed as a basic unit of automatic (computer) speech recognition as early as 1975 [7, 8], and this idea has been periodically reexamined (e.g. in [9, 10, 11, 12, 13]). The syllabic level confers several potential benefits; for one, syllabic boundaries are more precisely defined than phonetic segment boundaries in both the speech waveform and in spectrographic displays. Additionally, the syllable may serve as a natural organizational unit useful for reducing redundant computation and storage in decoding. The syllabic abstraction may also be appropriate for the incorporation of suprasegmental prosodic information.

English is considered to possess a highly complex syllabic structure not readily amenable to automatic segmentation or identification. Detailed statistical analyses of spontaneous informal discourse indicate that the syllabic structure of conversational English is not as complicated as has been generally supposed. For example, data gathered from telephone conversations in [14] and the Switchboard corpus [15, 16] indicate that over 80% of the word tokens in these corpora are monosyllabic, and more than 85% of the syllables are of the canonical consonant-vowel (CV), vowel-consonant (VC), V, or CVC varieties. These structural regularities can, in principle, be exploited to reliably estimate syllabic boundaries.

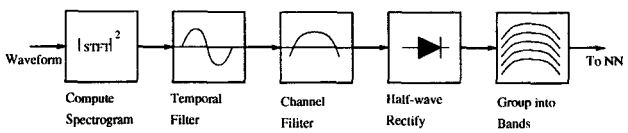

Figure 1. Major processing steps for the syllable onset features.

Previous research on detecting syllable boundaries and using this information to improve recognition accuracy is reported for English [8, 9, 10] and for German [12, 13]. In this communication we describe a perceptually-oriented method for the automatic delineation of syllabic onsets. Artificial neural networks (NNs) are used to classify both phonetic segments and potential syllabic onsets. In a departure from previous research, we focus on continuous, naturally-spoken English.

2. DETECTING SYLLABLE ONSETS

Syllable onsets are typically characterized by a pattern of synchronized rises in subband energy spanning adjacent subbands. The time course of these coordinated rises and falls in energy correspond to syllable-length intervals, on the order of 100-250 ms.

Figure 1 illustrates the signal processing procedures designed to enhance and extract these acoustic properties. The speech waveform is decomposed via short-time Fourier analysis into a narrow-band spectrogram, which is convolved with both a temporal and a channel filter, effectively creating a two-dimensional filter. The temporal filter (a high-pass filter analogous to a Gaussian derivative) smoothes and differentiates along the temporal axis, and is tuned for enhancing changes in energy on the order of 150 ms. The (Gaussian) channel filter performs a smoothing function across the channels, providing weight to regions of the spectrogram where adjacent channels are changing in coordinated fashion. Half-wave rectification is used to preserve the positive changes in energy, thus emphasizing the syllable onsets.

Large values in this representation correspond to positive-

0-8186-7919-0/97 $10.00

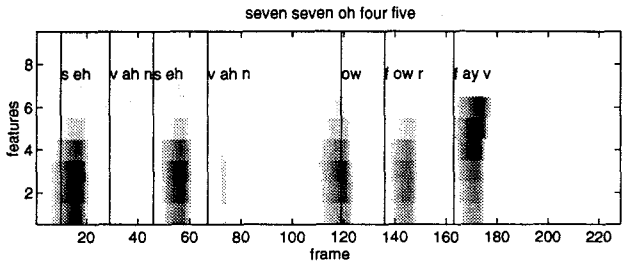

Figure 2. Example of onset features derived for the utterance 'seven seven oh four five'. The vertical lines denote syllable onsets as derived from hand-transcribed phone labels.

going energy regions where hypothesized syllable onset characteristics occur. The channel outputs are subsequently averaged over a region spanning nine critical bands [17], the result of which is illustrated in Figure 2.

Features derived from this procedure are updated every 10 ms. The resulting vectors are concatenated with log-RASTA [18] features computed over a 25-ms frame every 10 ms, and this combination is used as the input to a neural network classifier for estimating the location of syllabic onsets. A single-hidden layer, fully connected, feed-forward multilayer perceptron with 400 hidden units was trained to estimate the probability that a given frame is a syllable onset, given the acoustic patterns described above. For the purposes of training, the syllable onset (as derived from phonetically transcribed segmentation) was represented as a series of five frames, in which the initial frame corresponded to the actual onset.

A simple numeric threshold applied to the probability estimates generated by the neural net determined the identification of any given frame as a syllabic onset. This procedure correctly detected 94% of the onsets computed from phonetically transcribed data (within the five-frame tolerance window defined for training). The procedure also mistakenly inserted syllabic onsets where there were none (false positives) in 15% of the frames outside the tolerance window of any onset. These onset decisions were used by a syllable-based decoder as frames corresponding to syllable onsets.

3. SYLLABLE-BASED DECODING

A speech decoder was designed to incorporate an intermediate syllabic level of abstraction between the phone and word/sentence tiers. The decoder processes phonetic probabilities from a neural network using a conventional Viterbi algorithm using a bigram syllable grammar and creates a syllable graph (a derivative of the word graph as defined in [19]). The syllable graph serves as input to the program's stack decoder [20, 21], along with a bigram word grammar, to determine the most likely sequence of words. This procedure is a variation on the multiple-pass decoding method (related to the approach used in [22] and [23]) and enables the use of a complex language model at a higher stage of linguistic representation. The additional complexity of the decoder design permits the explicit representation of the relationship of phones to syllables and syllables to words. Syllabic onset information is introduced as an additional probability input into the decoder at the level of the syllable graph.

4. RECOGNITION EXPERIMENTS

Recognition experiments were performed on a subset of the OGI Numbers corpus [24]. This corpus contains continuous, naturally spoken utterances of many different speakers saying numbers from a vocabulary of thirty words. A sample utterance from the database is "eighteen thirty one." The example in Figure 2 is also derived from the Numbers corpus. Collected over telephone lines, the utterances exhibit large variations in speaking rate and acoustic environmental conditions. The subset includes approximately three hours (3500 sentences) of training data, and one hour (1200 sentences) each of development-test-set and final-test-set data. The training data, with its cross-validation subset, was used for tuning the parameters. The development test set (referred to as the "test set" in the sections below) was used for the results reported below.

4.1. Experiments with Syllabic Onsets Determined from Forced-Viterbi Alignment

In order to ascertain the potential value of syllabic onset timing, this information (derived from advance knowledge of the word-transcriptions of the test utterances) was incorporated into the decoding process.

A forced-Viterbi technique was used to generate phone alignment labels based on word transcriptions of the corpus provided for all the utterances in the test set. Artificial syllabic onsets were derived from these forced-Viterbi labels. The resulting syllabic onset information was only approximate. Many of the onsets were as much as 50 ms distant from the labelled segment boundary.

The experimental lexicon included 32 single-pronunciation words, comprising 30 different syllables. The pronunciations were derived from those developed at Carnegie Mellon University for large vocabulary recognition. The context-dependent phonetic durations used were derived from the training data using an embedded training process.

The recognition procedure used a highly restrictive criterion for syllabic decoding. A syllable was presumed to occur only when the beginning frame for the syllabic model coincided precisely with a predetermined onset. No restriction was placed on a syllable's termination; it was theoretically possible for the end point of a postulated syllable to occur after the next Viterbi-derived onset of the following syllable. Only syllabic onset information from the test set was included in our recognition experiments. No prior knowledge of phonetic information from the test set was used.

If the dynamic programming (Viterbi) procedure and the speech decoding input elements were of the ideal form, the addition of artificially-derived syllabic boundary information would, in theory, provide little or no improvement in recognition performance. In principle the decoding process assumes that models can begin at any frame, including the ones we specified as incorporating syllabic onsets. In our experiment, incorporation of artificially-derived syllable segmentation information reduces the word error rate by 38%, from 10.8% to 6.7%, as shown in Table 1. This large reduction in word error suggests that syllabic boundary information can significantly improve speech recognition performance when directly incorporated into the decoding process.

A second series of experiments was conducted with the aim of delineating the precision required for syllabic onset information to be of significant utility in the decoding process. The temporal precision of the syllabic onset was systematically varied over several frames, as shown for selected values in Table 2. There is a small, but significant

System	Word Error Rate sub./ins./del.
no onset information	10.8% 5.8%/3.1%/1.8%
with known syllable onset times Total frs./onset = 1	6.7% 4.4%/0.7%1.6%

Table 1. Word-error rates for decoding using a *single-pronunciation* lexicon, with and without artificial syllabic onsets derived from forced alignment.

Number of frames about each onset	Error Rate sub./ins./del.
Total frs./onset = 5 centered on onset	7.3% 4.9%/0.9%/1.5%
Total frs./onset = 9 centered on onset	7.8% 5.1%/1.3%/1.4%
Total frs./onset = 13 centered on onset	8.5% 5.2%/1.9%/1.4%

Table 2. Word-error rates for single-pronunciation decoding, using syllable hypotheses that are allowed to begin within several frames of artificial onsets derived from forced alignment.

increase in word error rate as the onset window is increased from one to thirteen frames, consistent with the hypothesis that syllabic onset information of intermediate accuracy is of potential utility in speech recognition systems.

4.2. Experiments with Acoustically Determined Syllabic Onsets

Speech recognition systems do not typically possess detailed *a priori* information concerning the temporal loci of syllabic boundaries. Rather, they must infer the syllable boundaries from other information sources. We are in the initial stages of integrating the acoustically-derived onset information described above into the decoding process.

In order to provide a closer match between the phonetically transcribed material and the syllabic onsets derived from the neural network training procedure, a new set of lexicons and grammars were developed, specifically based on the transcription data from the training set. These materials included 32 words (and their range of 178 possible pronunciations), comprising 118 separate syllabic forms. The spectrum of pronunciations included cover approximately 90% of the pronunciation variations in the corpus, as reflected in the phonetic transcription material. The durations of phonetic segments were also computed from the transcription of the training materials. The word grammar (derived from the word transcriptions of the training set) was identical to the one described for the initial series of recognition experiments in the last section. However, the syllable-level grammar was, by necessity, specifically adapted to this language model set.

The decoder used a simple threshold applied to the output of the neural network in order to ascertain the occurrence of a syllabic onset. The algorithm set temporal restrictions on the syllabic models such that they were required to begin no sooner than five frames preceding the time of the estimated syllabic onset. By this metric it was possible to reduce the number of potential starting frames for syllabic models by 58%.

System	Error Rate sub./ins./del.
with data-derived lexicon *no onset information*	9.1% 5.3%1.3%2.4%
with data-derived lexicon onset used with threshold only	8.2% 4.8%1.3%2.1%

Table 3. Word-error rates for multiple-pronunciation (data-derived) decoding, with and without *acoustically-derived* onsets.

When such acoustically-derived syllabic onset information is incorporated into the decoding process the recognition performance improves slightly. The word error rate decreases by 10% which, while not quite reaching statistical significance (for $p < 0.05$), is still indicative of the potential performance benefit to be derived from including temporal information pertaining to syllabic boundaries.

The incorporation of multiple pronunciations in the recognition lexicon improved the performance of the baseline system and served to provide further details concerning the specific relationship between syllabic boundary information and word models.

5. DISCUSSION

The experiments described in the section above illuminated certain limitations in the present recognition system that necessarily impact its performance. One such limitation of the current experimental paradigm pertains to the mismatch between the acoustic-phonetic and phonological representations of the syllable forms used for word recognition. The syllabic segmentation method was based largely on acoustic-phonetic criteria, while the syllabification of lexical items was derived from a more abstract phonological representation. An instance where this distinction is of particular significance for word sequences is one in which the syllable coda of the first word is consonantal and the onset of the following word is vocalic, as in "five eight." The phonological representation of such a sequence would be /fayv/ /eyt/, while the phonetic realization is more typically [fay] [veyt]. Such "re-syllabification" phenomena are not easily accommodated within the present syllabic representational framework. Future efforts will be devoted to resolving such issues within a single, coherent representational framework.

6. SUMMARY AND FUTURE WORK

Incorporation of information pertaining to syllabic onsets has the potential to significantly increase the accuracy of word-level recognition. This syllabic information was obtained in our study from two different sources – artificial boundaries derived from prior phonetic transcriptions of the corpus materials, and acoustic segmentation derived from a signal processing method based on general principles of auditory analysis. The word-error rate was reduced by 38% for the artificially-derived boundaries and by 10% for the boundary information derived from the acoustic segmentation method. These results indicate the potential utility of incorporating syllable boundary information in future speech recognition systems. We are now working towards improving the accuracy of the acoustically-based segmentation algorithm via the incorporation of the computed probability estimates from the neural net and through optimization of the decision criterion derived from such signal

detection theoretic parameters as the false alarm rate and response bias.

7. ACKNOWLEDGMENTS

We thank Dan Gildea for developing the data-derived pronunciations and gratefully acknowledge valuable assistance from Eric Fosler and Dan Ellis. The automatic syllabification program we used, *tsylb2*, was written by Bill Fisher of NIST.

This material is based upon research supported by the following funding sources: a National Science Foundation Graduate Research Fellowship (SW), Joint Services Electronics Program grant (SW, MS), Contract Number F49620-94-C-0038 and a DOD subcontract from the Oregon Graduate Institute. Additional support was received from the Faculté Polytechnique de Mons as part of a European Community Basic Research grant (Project Sprach). Finally, we gratefully acknowledge continued support from the International Computer Science Institute.

REFERENCES

[1] Antonio Bonafonte, Rafael Estany, and Eugenio Vives. Study of subword units for spanish speech recognition. In *Eurospeech*, volume 3, pages 1607–1610, Madrid, Spain, September 1995. ESCA.

[2] Sung-Chien Lin, Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee. A syllable-based very-large-vocabulary voice retrieval system for Chinese databases with textual attributes. In *Eurospeech*, volume 1, pages 203–206, Madrid, Spain, September 1995. ESCA.

[3] Dominic W. Massaro. Perceptual units in speech recognition. *Journal of Experimental Psychology*, 102(2):199–208, 1974.

[4] Douglas O'Shaughnessy. *Speech Communication*, chapter 5, pages 164–203. Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.

[5] Juan Segui, Emmanuel Dupoux, and Jacques Mehler. The role of the syllable in speech segmentation, phoneme identification and lexical access. In Gerry Altmann, editor, *Cognitive Models of Speech Processing*, chapter 12, pages 263–280. MIT Press, 1990.

[6] Steven Greenberg. Understanding speech understanding: Towards a unified theory of speech perception. In *Proceedings of the ESCA Workshop (ETRW) on The Auditory Basis of Speech Perception*, pages 1–8, Keele, United Kingdom, July 1996. ESCA.

[7] Osamu Fujimura. Syllable as a unit of speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(1):82–87, February 1975.

[8] Paul Mermelstein. Automatic segmentation of speech into syllabic units. *J. Acoust. Soc. Am*, 58(4):880–883, October 1975.

[9] M.J. Hunt, M. Lennig, and P. Mermelstein. Experiments in syllable-based recognition of continuous speech. In *ICASSP*, volume 3, pages 880–883, Denver, Colorado, April 1980. IEEE.

[10] P.D. Green, N. R. Kew, and D. A. Miller. Speech representations in the sylk recognition project. In M. P. Cooke, S. W. Beet, and M. D. Crawford, editors, *Visual Representation of Speech Signals*, chapter 26, pages 265–272. John Wiley, 1993.

[11] Kenneth W. Church. Phonological parsing and lexical retrieval. In Uli H. Frauenfelder and Lorraine Komisarjevsky Tyler, editors, *Spoken Word Recognition*, Cognition Special Issues, chapter 3, pages 53–69. MIT Press, 1987.

[12] W. Reichl and G. Ruske. Syllable segmentation of continuous speech with artificial neural networks. In *Eurospeech*, pages 1771–1774, Berlin, Germany, September 1993.

[13] Katrin Kirchhoff. Syllable-level desynchronisation of phonetic features for speech recognition. In *ICSLP*, volume 4, pages 2274–2276, Philadephia, Pennsylvania, October 1996.

[14] Norman R. French, Charles W. Carter, Jr., and Walter Koenig, Jr. The words and sounds of telephone conversations. *The Bell System Technical Journal*, IX:290–325, April 1930.

[15] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *ICASSP*, volume 1, pages 517–520, San Francisco, California, March 1992. IEEE.

[16] Steven Greenberg, Joy Hollenback, and Dan Ellis. The Switchboard transcription project. Technical report, International Computer Science Institute, 1997.

[17] Donald D. Greenwood. Critical bandwidth and the frequency coordinates of the basilar membrane. *JASA*, 33:1344–1356, 1961.

[18] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, October 1994.

[19] Martin Oerder and Hermann Ney. Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *ICASSP*, volume 2, pages 119–122, Minneapolis, Minnesota, April 1993. IEEE.

[20] Frederick Jelinek. Fast sequential decoding algorithm using a stack. *IBM J. Res. Develop.*, 13:675–685, November 1969.

[21] Steve Renals and Mike Hochberg. Efficient evaluation of the LVCSR search space using the noway decoder. In *ICASSP*, volume 1, pages 149–152, Atlanta, Georgia, May 1996. IEEE.

[22] Frank K. Soong and Eng-Fong Huang. A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition. In *ICASSP*, volume 1, pages 705–708, Toronto, Canada, May 1991. IEEE.

[23] P. Kenny, R. Hollan, V. Gupta, M Lennig, P Mermelstein, and D. O'Shaughnessy. A^*-admissible heuristics for rapid lexical access. In *ICASSP*, volume 1, pages 689–692, Toronto, Canada, May 1991. IEEE.

[24] Center for Spoken Language Understanding, Department of Computer Science and Engineering, Oregon Graduate Institute. Numbers corpus, release 1.0, 1995.

[25] Godfrey Dewey. *Relative Frequency of English Speech Sounds*, volume 4 of *Harvard Studies in Education*. Harvard University Press, Cambridge, 1923.

[26] Zhihong Hu, Johan Schalkwyk, Etienne Barnard, and Ronald Cole. Speech recognition using syllable-like units. In *ICSLP*, volume 2, pages 1117–1120, Philadephia, Pennsylvania, October 1996.