Experiments in Lexical Access for Speech Recognition

David Albrecht and Ingrid Zukerman
Faculty of Information Technology
Monash University, Clayton, VIC 3800, Australia

Ian Thomas*
Department of Computer Science and Information Technology
RMIT, 124 La Trobe Street, VIC 3001, Australia

Abstract

The *Lexical access problem* consists of determining the sequence of words that corresponds to a spoken utterance. In this paper, we present a Source-Channel model for addressing the lexical access problem, and describe experiments which investigate the impact of different choices regarding modeling and search parameters on performance. Our results show that the following yield significant improvements in lexical access performance: (1) using Dirichlet priors for estimating the back-off factor employed in a deleted-interpolation smoothing model, and (2) performing regional word adjustments during a post-processing stage. In addition, the use of short-lists of candidate words based on acoustic similarity significantly speeds up performance, with only a small drop in accuracy.

1 Introduction

The *Lexical Access Problem* consists of determining the sequence of words that corresponds to a spoken utterance. This problem is fraught with uncertainty due to the mismatch between the wave patterns corresponding to lexicon words and those received from a speaker, and the difficulties associated with word boundary detection. In this paper, we report on the results obtained from an empirical study involving key language modeling and search parameters. Specifically, our contribution pertains to the effect of these parameters and the interactions between them on system performance.

Our general approach to the lexical access problem follows the commonly used hypothesize-and-test paradigm [44, 22, 37, 11]. Hypothesis generation is done by means of a search procedure, and hypothesis evaluation by a model that estimates the probability that a hypothesized sentence is indeed the uttered sentence. The sentence with the highest probability is that returned by the system.

*The research described in this paper was conducted while Ian Thomas was a PhD student at Monash University.

Hypothesis generation is performed by a standard expansion-selection process, e.g., [32]. In most speech recognition systems, lexical access is integrated with phoneme recognition and language modeling in the form of a graph search through a large Hidden Markov Model (HMM), e.g., [15, 20, 7]. However, many research systems [4, 39, 43, 13, 26] have a more modular structure, which allows different components to be analyzed and evaluated separately. We follow this approach by separating the high-level components for word-hypothesis generation and language modeling from the lower-level acoustic and phonetic modeling components. That is, we investigate lexical access from a sequence of phonetic symbols, rather than from a speech wave. The reliance on phonetic symbols has two opposite effects. On one hand, these symbols are the best phonetic candidates produced by a phoneme recognizer that works directly on the speech signal [17, 27]. On the other hand, working only with phonetic symbols means that we cannot make hypotheses regarding word boundaries, which are informed by acoustic features of the speech signal [31, 24].

As a result of these effects, the search space is still potentially huge. In order to reduce the number of candidate words, we have considered heuristics that cluster words into equivalence classes, and retain only promising candidates from these classes. Clearly, the resultant reduction in the search space may also result in loss of accuracy. To alleviate this problem, we have investigated a post-processing step, where potentially erroneous parts of the top-ranked sentence hypothesis are identified and replaced if necessary.

Hypothesis evaluation is done by means of a Source Channel Model (SCM) [8, 36], where an intended English sentence is transmitted through a noisy channel (speech) to the machine, which then produces an interpretation. Ideally, this interpretation should match the intended sentence. However, inaccuracies accumulated during the transmission process may cause differences between the intended sentence and that determined by the receiver. The SCM estimates the probability that each candidate sentence postulated by the receiver is the intended sentence. Our model is broken down into a language model and a word model.

- The language model describes words in the context of a sentence. It estimates the probability of language events using a mixture of second and third order Markov chains.
- The word model describes each hypothesized word in a sentence in terms of its component phonetic symbols. It estimates the probability of a hypothesized word using an edit distance algorithm which calculates the similarity between the phonetic sequence corresponding to this word and an input phonetic subsequence that has been proposed as a word.

The modeling and search parameters investigated in our empirical study are:

- Two competing approaches for estimating the back-off factor used in a deleted-interpolation smoothing model employed to address sparse data problem during language modeling (Section 2.1).
- Two competing approaches for calculating the probability of a particular word during word modeling (Section 2.2).

- Two complementary methods for reducing the number of word candidates being considered during search:
  - generating a short-list of likely possibilities (Section 3.1); and
  - pruning this list after it has been evaluated by the word model, so that only the most promising candidates are considered by the language model — we considered two competing approaches for this method (Section 3.2).
- Performing regional word adjustments during a post-processing step (Section 3.3).

This paper is organized as follows. In Section 2, we describe our sentence evaluation method, and our approaches for calculating the smoothing factor and the probability of a word realization. Section 3 details the procedure for searching through candidate sentences given an input phonetic sequence, and discusses our heuristics for expediting the search process. Our results are presented in Section 4. Related work is discussed in Section 5, and concluding remarks in Section 6.

2 The Sentence Model

We define a sentence S as a sequence of words $w_1, \ldots, w_n$, together with their corresponding parts-of-speech $\mathrm{PoS}_1, \ldots, \mathrm{PoS}_n$, and partition the input phonetic sequence into n segments $phs_1, \ldots, phs_n$, where segment phs_i is the sequence of phonetic symbols corresponding to word w_i. The lexical access problem can then be stated as follows: determine the sentence S that is most probable, given the sequence of phonetic segments $phs_1, \ldots, phs_n$. That is,

$$S_{\max} = \mathrm{argmax}_S \Pr(S | phs_1, \ldots, phs_n)$$

By expanding S and applying Bayes rule we obtain

$$\begin{aligned} S_{\max} &= \mathrm{argmax}_S \{\Pr(w_1, \mathrm{PoS}_1, \ldots, w_n, \mathrm{PoS}_n | phs_1, \ldots, phs_n)\} \quad (1) \\ &= \mathrm{argmax}_S \{\Pr(phs_1, \ldots, phs_n | w_1, \mathrm{PoS}_1, \ldots, w_n, \mathrm{PoS}_n) \times \\ &\qquad \Pr(w_1, \mathrm{PoS}_1, \ldots, w_n, \mathrm{PoS}_n)\} \\ &= \mathrm{argmax}_S \left\{ \prod_{i=1}^{n} \Pr(phs_i | w_1, \mathrm{PoS}_1, \ldots, w_n, \mathrm{PoS}_n, phs_1, \ldots, phs_{i-1}) \times \right. \\ &\qquad \left. \prod_{i=1}^{n} \Pr(w_i, \mathrm{PoS}_i | w_1, \mathrm{PoS}_1, \ldots, w_{i-1}, \mathrm{PoS}_{i-1}) \right\} \end{aligned}$$

The first factor represents the word model, and the second factor the language model. In order to obtain reliable estimates of these factors, we make the following simplifying assumptions: (1) the probability of the current word w_i depends only on the current part-of-speech PoS_i and the previous word w_{i-1}, (2) the probability of the current part-of-speech PoS_i depends only on the previous parts-of-speech PoS_{i-1} and

Table 1: The Wiretap corpus and the subset used for training the language model

Total number of words before matching with TIMIT	9.7M
Total number of words after matching with TIMIT	5.6M
Number of distinct words before matching with TIMIT	137,281
Number of distinct words after matching with TIMIT	5,753
Distinct words common to both Wiretap and TIMIT training set	4,583

PoS_{i-2}, and (3) the probability of a phonetic sequence phs_i depends only on the corresponding word w_i. This yields the following formulation for Equation 1.

$$S_{\max} = \text{argmax}_S \underbrace{\left\{ \prod_{i=1}^{n} \Pr(phs_i|w_i) \right.}_{\textit{Word Model}} \times \underbrace{\left. \prod_{i=1}^{n} \{\Pr(w_i|\text{PoS}_i, w_{i-1}) \times \Pr(\text{PoS}_i|\text{PoS}_{i-1}, \text{PoS}_{i-2})\} \right\}}_{\textit{Language Model}} \tag{2}$$

2.1 The Language Model

The language model represents the actual words that make up a sentence. Since a "sensible" sounding sentence is more desirable than a sentence composed of unrelated words, we require a set of training sentences that accurately describes sensible sentences. Furthermore, we require a model that can express the attributes of sensible sentences, and we need to estimate the probability of the values of these attributes. To represent sensible English sentences, we take into account the syntactic role of the words in these sentences, as well as the actual usage of these words. The former is done by preferring frequent part-of-speech combinations, e.g., an article followed by a noun, to infrequent ones, e.g., an article followed by another article; and the latter by preferring common word combinations.

Let $\Pr(\mathbf{W}, \mathbf{PoS})$ denote the Language Model portion of Equation 2, i.e., the probability of a word sequence $\mathbf{W}$ together with the corresponding parts-of-speech $\mathbf{PoS}$.

$$\Pr(\mathbf{W}, \mathbf{PoS}) = \prod_{i=1}^{n} \{\Pr(w_i|\text{PoS}_i, w_{i-1}) \times \Pr(\text{PoS}_i|\text{PoS}_{i-2}, \text{PoS}_{i-1})\} \tag{3}$$

We used a set of classic texts drawn from the Wiretap corpus (`http://wiretap.area.com`) to train our language model.[1] However, since the TIMIT corpus was used to train the word model (Section 2.2), the vocabulary of the Wiretap corpus was brought in line with TIMIT's vocabulary by retaining only those N-grams involving words in the TIMIT lexicon. Table 1 describes statistics from the Wiretap corpus.

As seen in Equation 3, to train the language model we need to estimate the probabilities $\Pr(w_i|\text{PoS}_i, w_{i-1})$ and $\Pr(\text{PoS}_i|\text{PoS}_{i-1}, \text{PoS}_{i-2})$. One approach to estimate these

[1] These publicly available texts include works such as Melville's *Moby Dick* and Bronte's *Wuthering Heights*.

probabilities consists of using Maximum Likelihood Estimates:

$$\widehat{\Pr}(w_i|\text{PoS}_i, w_{i-1}) = \frac{\text{Freq}(w_{i-1}, \text{PoS}_i, w_i)}{\text{Freq}(w_{i-1}, \text{PoS}_i)} \quad (4)$$

and

$$\widehat{\Pr}(\text{PoS}_i|\text{PoS}_{i-2}, \text{PoS}_{i-1}) = \frac{\text{Freq}(\text{PoS}_{i-2}, \text{PoS}_{i-1}, \text{PoS}_i)}{\text{Freq}(\text{PoS}_{i-2}, \text{PoS}_{i-1})} \quad (5)$$

However, these estimates may lead to *sparse data problems*. This happens when a configuration in the numerator or denominator of Equations 4 or 5 is encountered during testing, but was never observed in the training corpus (i.e., it has zero frequency). There are several solutions to the sparse data problem [21, 23, 2, 54, 34, 9]. In this paper, we adopt the approach known as *blended context model* or *deleted interpolation model* [21, 54], where the estimate of $\Pr(w_i|\text{PoS}_i, w_{i-1})$ is given by

$$\widehat{\Pr}(w_i|\text{PoS}_i, w_{i-1}) = (1-\lambda(w_i))\widehat{\Pr}_{PoS,W}(w_i) + \lambda(w_i)\widehat{\Pr}_{PoS}(w_i) \quad (6)$$

where

$$\widehat{\Pr}_{PoS,W}(w_i) = \frac{\text{Freq}(w_i, \text{PoS}_i, w_{i-1}) + \alpha}{\sum_w(\text{Freq}(w, \text{PoS}_i, w_{i-1}) + \alpha)} \quad (7)$$

$$\widehat{\Pr}_{PoS}(w_i) = \frac{\text{Freq}(w_i, \text{PoS}_i) + \beta}{\sum_w(\text{Freq}(w, \text{PoS}_i) + \beta)} \quad (8)$$

$$\lambda(w_i) = \frac{\gamma}{\sum_w(\text{Freq}(w, \text{PoS}_i, w_{i-1}) + \alpha) + \gamma} \quad (9)$$

The parameters α and β in Equations 7 and 8 are known as *Good's flattening constant* [16], and are used to avoid uncommon words being assigned a zero probability. In most of our experiments, we set α to 0.5 and β to 1 (Section 4). The idea is that higher flattening constants are suitable for coarser groupings (which have more occurrences of the values in question). The parameter γ in Equation 9 is used determine the probability that we need to back-off from the model that uses the context PoS_i, w_{i-1} to a simpler model that requires only context PoS_i. We considered two methods for determining the value of γ on the basis of the prior probability distribution of words given the context: *Uniform* and *Dirichlet* (Appendix A).

2.2 The Word Model

The word model describes how well the phonetic symbols uttered by the speaker match the words postulated by the search mechanism (Section 3). That is, the word model calculates the second factor of Equation 2, $\prod_{i=1}^{n}\Pr(phs_i|w_i)$.

Since a word may have several phonetic realizations, we consider two ways to calculate $\Pr(phs_i|w_i)$: *Max* and *Weighted*.

- *Max* uses $real_{i,best}$, the realization of w_i that best matches phs_i, yielding

$$\Pr(phs_i|w_i) = \Pr(phs_i|real_{i,best}) \times \Pr(real_{i,best}|w_i) \quad (10)$$

Table 2: The training and test subsets of the TIMIT corpus

TIMIT training set		Core TIMIT test set	
Total number of sentences	3,696	Total number of sentences	192
Total number of words	29,997	Total number of words	1,565
Distinct words	4,910	Distinct words	912
Total number of distinct words in TIMIT lexicon			6,224

where

$$real_{i,best} = \text{argmax}_{j=1}^{N_i}\{\Pr(phs_i|real_{ij})\}$$

N_i is the number of realizations of w_i, and $real_{ij}$ is the jth realization of word w_i.[2]

- *Weighted* uses all the realizations of w_i, weighing their contribution to $\Pr(phs_i|w_i)$ in proportion of their probability. This method yields

$$\Pr(phs_i|w_i) = \sum_{j=1}^{N_i}\{\Pr(phs_i|real_{ij}) \times \Pr(real_{ij}|w_i)\} \quad (11)$$

We used a portion of the TIMIT corpus [10, 14] to train our word model and test system performance. The TIMIT corpus is a collection of American-English read sentences with correct time-aligned acoustic-phonetic and orthographic (word-aligned) transcriptions. The entire dataset contains 6,300 sentences spoken by 630 speakers from 8 different dialect divisions across the United States. Each speaker says five phonetically-compact sentences and three phonetically-diverse sentences to give a good coverage of the phonemes in the language, and two dialectically diverse sentences designed to expose dialectic variations in the speakers. The sentences were recorded using a high-quality, headset-mounted microphone in a noise-isolated room, and speakers were instructed to read prompts in a "natural" voice. Our training set comprised 3,696 sentences and our test set comprised the 192 sentences in the TIMIT Core Test Set (we did not use the dialectically diverse sentences, as recommended in the TIMIT documentation [14]). Table 2 describes statistics from the TIMIT corpus.[3]

$\Pr(real_{ij}|w_i)$ – the probability of realization j of lexicon word w_i (the second factor in Equations 10 and 11), is obtained directly from the corpus. For example, Table 3 shows the frequencies for the realizations found in the corpus for the word "another",

[2] Another version of the *Max* method would select the realization $real_{ij}$ for which $\text{argmax}_{j=1}^{N_i}\{\Pr(phs_i|real_{ij}) \times \Pr(real_{ij}|w_i)\}$. We did not implement this option, as it de-emphasizes the goodness of a phonetic match.

[3] The lexicon has more words than the sum of the distinct words in the training and test sets, because we used only part of the corpus for training and testing.

Table 3: Phoneme realizations for the word another with frequencies of occurrence in the training corpus

Realization	# of occurrences in training set	Phonetic symbol sequence					
another$_1$	3	ix	n	ah	dh	axr	
another$_2$	3	ax	n	ah	dh	er	
another$_3$	2	q	ax	n	ah	dh	axr
another$_4$	1	q	ax	n	ah	dh	er
another$_5$	1	q	ax	n	ah	dh	ax
another$_6$	1	ix	nx	ah	dh	uh	
another$_7$	1	er	n	ah	dh	axr	

yielding the following probability for another$_2$.

$$\Pr(\texttt{another}_2|\text{another}) = \frac{\text{Frequency of } \texttt{another}_2}{\sum_i \text{Frequency of } \texttt{another}_i} = \frac{3}{12}$$

$\Pr(\mathbf{phs_i}|\mathbf{real_{ij}})$ – the probability of an input phonetic sequence phs_i given the phonetic sequence for a particular realization of a word (the first factor in Equations 10 and 11), is calculated on the basis of the operations (insertions, deletions and substitutions) required to transform a particular realization of a postulated word into the actual input phonetic symbols. For example, suppose that the language model has postulated the word "another" as a candidate for the phonetic symbol sequence ax n dx q er. As seen in Table 3, this sequence does not correspond exactly to any of the realizations of "another". The realization that best matches the input is another$_2$, yielding the following alignment.

another$_2$	ax	n	ah	dh	--	er
input	ax	n	--	dx	q	er

For these sequences to match, we must delete ah from another$_2$, substitute dh with dx, and insert q.

We apply an edit distance algorithm [48, 33, 42] to determine the optimal (highest probability) alignment between an input phonetic sequence and a realization. The algorithm takes into account the probabilities of phoneme insertions, deletions and substitutions, yielding the following probability for an alignment between a phonetic sequence phs_i and a realization $real_{ij}$.

$$\Pr(phs_i|real_{ij}) = \underbrace{\Pr(N \text{ insertions in } real_{ij}) \times \frac{1}{\binom{L+N}{N}}}_{Insertions} \times \underbrace{\prod_{k=1}^{L+N} \Pr(\text{substitution } k)}_{Substitutions} \quad (12)$$

where N is the number of phonemes inserted in $real_{ij}$, and L is the number of phonemes that were originally in $real_{ij}$. The first two factors in Equation 12 represent insertions, and the third factor represents substitutions.

- Insertions are treated differently from substitutions and deletions, because they increase the number of operations in the alignment between an input phonetic sequence and a realization, and require indicators that specify the position of inserted phonemes in the alignment.

  – The first factor in Equation 12, which represents the probability of inserting N phonemes in realization $real_{ij}$, is obtained from training data as follows. 10% of the database of realizations seen in training are removed; each realization in this subset is optimally aligned with the remaining realizations for the same word in the database; and the number of insertions in the closest alignment is recorded. This process is repeated with a different 10% of the database, until all phonetic realizations in the original database have been inspected once.

  – The second factor in Equation 12, which represents the probability of inserting N phonemes in particular positions in an alignment of length $L+N$ (the length of the realization plus the number of insertions) is obtained from combinatorics. In the above example, only one insertion was performed — `q` between `dx` and `er` — yielding an alignment of length 6 (5 phonetic symbols in the chosen lexical realization plus 1 insertion). The probability of choosing a particular position for this insertion is $\frac{1}{6}$.

- Once the positions of insertions have been specified, all the operations are considered substitutions (deletions are substitutions of "$-$" for a phoneme, and insertions are substitutions of a phoneme for "$-$"). In our example, the third factor in Equation 12 is calculated by multiplying the probabilities Pr(`ax`|`ax`), Pr(`n`|`n`), Pr($-$|`ah`), Pr(`dx`|`dh`), Pr(`q`|$-$) and Pr(`er`|`er`). These probabilities are obtained from training data as follows. First, for each word in the lexicon, we obtain from the corpus a set of possible phonetic realizations. Next, we align the canonical realization of each lexicon word with its realization in the corpus, using high-certainty alignments as anchors, and iteratively reducing the low-certainty areas. Figure 1 shows an example of a TIMIT sentence where the canonical phonetic symbols from words in the lexicon (second row) are aligned with actual phonetic symbols spoken by one of the TIMIT speakers (third row). First, we align the identical phonetic symbols (e.g., `r`, `n` and `z` in `ruins`), next we align different symbols that are flanked by high-certainty symbols (e.g., `ih` and `ix` in `neoclassic`). The effect of these new alignments often reduces larger areas of uncertainty (e.g., the `ih/ix` match in `neoclassic` has this effect on the {`uw/ux` `ih/ix`} area in `ruins`, yielding a high-certainty match for `uw/ux`). This process returns the frequency for all possible matches of phonetic symbols. Exact matches have a high probability, phonetically similar substitutions, such as `dh` and `dx`, have a lower but still substantial probability, and phonetically dissimilar substitutions have the lowest probability [47, 46].

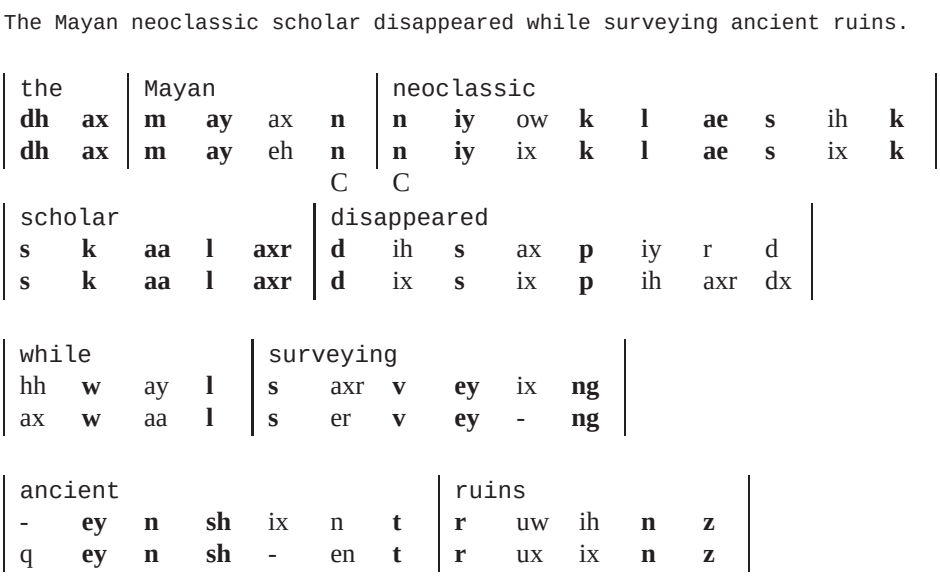

Figure 1: An alignment of the canonical phonetic realizations for the words of a sentence with the input phonetic sequence for that sentence

3 Search Through Possible Sentences

Efforts in lexical access have often followed the hypothesise-and-test paradigm, where the waveform corresponding to a word is partitioned into distinct segments, and each segment is labelled according to relatively reliable information extracted from the waveform (such as whether the segment is voiced or unvoiced) [44, 22, 37, 11]. Word candidates with label representations that closely match the postulated string of labels become hypotheses for further consideration. A more detailed and time-consuming analysis is carried out to determine which of these words best represents the underlying waveform [12]. We use a similar method, but with respect to phonetic symbols, rather than waveform labels.

We have implemented a modified version of the level-building algorithm, which proposes word boundaries, and expands partial sentence hypotheses a word at a time [32]. In our case, an expansion consists of first using the word model to match words from the lexicon to the postulated sequences of phonetic symbols (Section 2.2). These words are then assigned candidate part-of-speech tags, and new partial sentences are generated by appending each tagged word to the previous partial sentences. These new partial sentences are evaluated by the sentence model, combining the results of the language model with those of the word model (Section 2).

As described in Section 2.2, the word model estimates the probability of a hypothesized word by finding the optimal alignment between its phonetic realizations and the input phonetic symbols. This is a time consuming process if all the words in a reasonably-sized lexicon (comprising thousands of words) are considered. To expedite

this process, we investigate two complementary approaches for reducing the number of word candidates to be evaluated: (1) generating a short-list of likely possibilities (Section 3.1); and (2) pruning this list after it has been evaluated by the word model, so that only the most promising candidates are evaluated by the language model (Section 3.2).

3.1 The Short List

Arising from research into auditory word models and the structure of lexicons, the *Cohort theory* or *Phonetic Refinement Process* examines an input signal corresponding to a word in progressively more detail, rather than examining the input signal in one single detailed analysis stage [28, 37, 35, 45]. Those candidates that have a low score at an early stage are immediately eliminated, and do not enter the next stage. Each stage classifies an input signal into a class from a set of *equivalence classes*. These classes are defined by the values of features obtained from input signals — each class contains a list of words that have similar values for these features. Features that define the set of equivalence classes are more robust (undergo less variation due to the presence of noise) than the features that are used to compare individual word candidates to a signal. Given an input signal, the system chooses an equivalence class of words which have similar features to those of the input signal, removing from consideration words from other equivalence classes (whose features are significantly different from the features of the input signal). This pre-classification stage significantly reduces the number of candidate words considered in the later, intensive word analysis stage.

The phonetic refinement process may be applied in a variety of ways. It can be used to either make a firm hypothesis for a word [41], or reduce the number of hypotheses. Huttenlocher and Zue [19] describe a system that determines the manner of articulation (vowel, fricative, stop, etc) for segments of the signal for an isolated word, and searches the lexicon for words that have the same articulation pattern. The manner of articulation, denoted *Broad Sound Group (BSG)*, may be extracted more reliably than phonemes, and can be used to score and eliminate word hypotheses. Huttenlocher and Zue show that even if certain parts of the utterance segment are ignored (such as lexically unstressed syllables), the stressed syllables carry enough information to reduce substantially the number of word candidates that need to be considered. Waibel [49] uses separate subsystems, each of which individually searches the lexicon for words that match the pattern of one feature extracted from the input utterance; he considers segment durations, intensity, and stress as features. Shipman and Zue [44] perform a similar operation with vowel and consonant sequences. Fissore *et al.* [12] and Chen [5] express the word pronunciation network as a tree that is searched to derive word candidates under phonological and acoustic constraints. Finally, Tang *et al.* [45] model the manner and articulation of speech features as a pre-classification step prior to detailed acoustic and phonetic analysis.

We cluster the realizations of training words into equivalence classes (Section 3.1.1). Following [19], this clustering process uses BSGs as the basis for the features of the words. During recognition, the following procedure is used to exploit these equivalence classes.

1. Map the phonetic sequence corresponding to an input word to a set of feature

Table 4: The Broad Sound Group (BSG) classification of the TIMIT phonetic symbols

Stop	b, d, g, p, q, t, k, dx
Affricate	jh, ch
Fricative	z, zh, v, dh, s, sh, f, v, th
Nasal	m, n, nx, ng, em, en, eng
Semi-vowel/Glide	l, r, y, w, el, hh, hv
Vowel	iy, ih, eh, ae, aa, er, ah, ax, ao, uw, uh, ow, axr, ax-h
Pauses	pau, epi
Word Marker	endword

values based on BSGs.

2. Determine the class *C* which contains words that are most similar to the input phonetic sequence, i.e., the class to which this sequence has the highest probability of belonging (Section 3.1.2).

3. Use the word model to evaluate each word in class *C* as a word hypothesis for the input phonetic sequence.

4. Select the highest scoring hypotheses for evaluation by the language model (Section 3.2).

3.1.1 Determining Word Equivalence Classes

As mentioned above, we use BSGs as the basis for the clustering and classification of phonetic sequences. Recognition of BSG symbols is more reliable than recognition of individual phonetic symbols, because BSG symbols correspond to distinctive classes of articulation, and generally have strong correlating features in the signal [38]. Furthermore, a substitution error in a phonetic symbol is likely to maintain the same BSG symbol [47]. Hence, an input word with a phonetic sequence that has an erroneous symbol is still likely to be placed in the same equivalence class as the intended word. A more precise second stage of verification is then used to uncover the intended word corresponding to an input phonetic sequence (in our case, the evaluation of the input phonetic sequence by the word model, and then the sentence model). In a complete speech recognition system, the BSG symbols would be determined by acoustic analysis of the signal. In our experiments, the BSG sequence is determined from the input phonetic sequence by using Table 4.

At first glance, one would think of setting up each equivalence class to contain words that have the same sequence of BSG symbols. Thus, if one or more input phonetic symbols were changed by mis-recognition or mispronunciation (for example, an accent causing a vowel to be realized as a different vowel), then the input phonetic sequence would still map onto the same BSG sequence, and therefore would remain in the same equivalence class. However, if sounds were added by noise or removed by co-articulation, then the length of the BSG sequence would change, and a different

equivalence class would be selected. Hence, such a sequence-based scheme would be intolerant to noise due to insertions and deletions.

To overcome this problem, we create word equivalence classes by applying a clustering process which uses features that are independent of the length of a phonetic sequence (but their values depend on the length of this sequence). The *BSG bigram count* scheme, which yielded the best performance among those we investigated, encodes the features of a word as the frequencies of all possible pairs of BSG symbols.[4] Thus, each word is represented by 64 features (there are seven BSG symbols plus an end-of-word marker). For example, the set of nonzero features for the word `neoclassic` is `1 nasal/vowel`, `1 vowel/vowel`, `2 vowel/stop`, `1 stop/glide`, `1 glide/vowel`, `1 vowel/fricative`, `1 fricative/vowel`, and `1 stop/endword`.

We employed the Snob algorithm [51, 50] to generate equivalence classes from the phonetic sequences corresponding to the 29,997 words in the TIMIT training sentences (Table 2). These sequences cover all the phonetic variants of each distinct word in the training corpus. Using the BSG bigram count scheme, Snob generated 38 equivalence classes, with a mean class size of 496 words and a standard deviation of 337 words (different variants of a word may appear in different equivalence classes). Table 5 shows examples of words in some of these equivalence classes.

3.1.2 Phonetic Sequence Classification using Word Equivalence Classes

Once a set of classes has been generated by Snob, we classify a phonetic sequence phs_i corresponding to a new word w_i into the class that hopefully contains the intended word. To this effect, we first map phs_i to our set of features (BSG bigram counts). A Naïve Bayes Classifier [29] is then employed to assign phs_i to the most probable class $C_{\max}$ for this set of features. That is,

$$C_{\max} = \text{argmax}_{c=0,\ldots,\mathscr{C}-1} \{\Pr(\mathbf{F}_i|c) \times \Pr(c)\} \tag{13}$$

where $\mathscr{C}$ is the number of classes, and $\mathbf{F}_i$ is a 64-dimensional vector of BSG bigram counts for phs_i.

Since a Naïve Bayes Classifier assumes conditional independence of the individual features, we obtain

$$C_{\max} = \text{argmax}_{c=0,\ldots,\mathscr{C}-1} \left\{ \prod_{j=1}^{N} \Pr(\mathbf{F}_i(j)|c) \times \Pr(c) \right\} \tag{14}$$

where $N = 64$, and $\Pr(\mathbf{F}_i(j)|c)$ is the probability of BSG bigram j appearing $\mathbf{F}_i(j)$ times in the words in class c (this probability is returned by Snob). For example, if the input phonetic sequence had 2 `vowel/stop`s, $\Pr(2$ `vowel/stop`$|c)$ would be obtained from the distribution of `vowel/stop` bigrams in the words in class c.

Using the classes produced with the BSG bigram count scheme, we found that 94% of the 1,565 TIMIT test words were classified into the correct class, where a class is deemed correct if at least one realization of the intended word appears in that class.

[4] The other schemes were *BSG count*, which encodes the frequencies of individual BSG symbols; and *BSG count + length of phonetic sequence*.

Table 5: Sample words in equivalence classes for the BSG bigram count scheme

Class 0	Class 1	Class 2	Class 3	...
me	odds	emerge	of	
more	apple	image	odd	
my	eggs	enough	had	
now	echo	amaze	said	
knee	certain	advance	set	
the	upper	unless	that	
to	occur	enamel	at	
not	outer	agenda	egg	
know	atoms	engulfed	step	
meet	aloud	efficient	sought	
mind	again	inaugural	ought	
mine	effect	attendants	act	
nine	awkward	equipment	urged	
memo	obtain	accordingly	food	
number	earthquake	remembered	aid	
instead	advance	eyestrain	eight	
moment	overweight	amounts	out	
mechanism	occupied	unpleasant	it	
experiment	eruption	interesting	oak	
extraordinary	unbeatable	underwriting	or	
⋮	⋮	⋮	⋮	

This level of performance may be attributed to the ability of this scheme to represent some positional context for phonetic symbols. It is also worth noting that the outcome of a classification into the wrong class may still be corrected during the final adjustment stage (Section 3.3).

3.2 Pruning the Word Hypothesis List

We have described a method that reduces the number of word candidates to be evaluated by the word model. The word model then calculates the probability of the match between an input phonetic sequence and each word in the chosen equivalence class (Section 3.1.2). We now describe a pruning process that reduces the number of word candidates that must be considered by the language model when forming new partial sentence hypotheses. This process requires a metric for ranking the word candidates, and a threshold for pruning the unpromising candidates.

A metric for ranking word hypotheses. As described in Section 3.1.2, the word model produces a set of L word hypotheses $\{w_i(0), w_i(1), \ldots, w_i(L-1)\}$ for the input phonetic sequence phs_i corresponding to word w_i in a sentence. We use $\Pr(w_i(j)|phs_i)$,

Table 6: A subset of the word candidate list for the phonetic sequence `hv ix m s` ranked by $\Pr(w_i(j)|phs_i)$

Rank	$w_i(j)$	Freq$(w_i(j))$	$\Pr(phs_i\|w_i(j))$	$\Pr(w_i(j)\|phs_i)$
0	`house`	7,555	2.33×10^{-5}	0.402
1	`hands`	6,166	2.18×10^{-5}	0.307
2	`homes`	235	4.92×10^{-4}	0.264
3	`harms`	12	9.81×10^{-6}	0.000269
...	...	...	...	...
192	`lightbulbs`	1	3.25×10^{-16}	7.42×10^{-16}

the probability of word $w_i(j)$ given the phonetic sequence phs_i, to rank these hypotheses.

$$\Pr(w_i(j)|phs_i) = \frac{\Pr(phs_i|w_i(j))\ \Pr(w_i(j))}{\sum_{k=0}^{L-1} \{\Pr(phs_i|w_i(k))\ \Pr(w_i(k))\}}$$

where $\Pr(phs_i|w_i(j))$ is obtained from Equation 12 (Section 2.2), and $\Pr(w_i(j))$ is calculated using Maximum Likelihood Estimation on the basis of word frequencies in the training corpus. The word hypotheses are then ranked in descending order of probability, i.e., $w_i(0)$ has the highest probability.

A threshold for pruning word candidates. This threshold should depend on the relative merit of the candidate hypotheses. That is, if there is little variation between the probabilities of the top ranked word hypotheses, then all these hypotheses should be retained. However, if there are some clearly superior word candidates, then only these candidates should be retained. At the same time, the threshold should retain word hypotheses with relatively low probability, but which have a chance of redeeming themselves when evaluated by the language model.

These considerations prompted us to use a threshold that supports the selection of a variable number of candidates depending on their relative quality. Specifically, we retain a word hypothesis if its probability is greater than a proportion T of the probability of the best word $w_i(0)$ in the ranked set of words. That is, a word hypothesis $w_i(j)$ is retained if

$$\Pr(w_i(j)|phs_i) > T \times \Pr(w_i(0)|phs_i) \tag{15}$$

Table 6 shows a small subset of the 193 word candidates for the phonetic sequence `hv ix m s` ranked in descending order of the probability $\Pr(w_i(j)|phs_i)$. The word `homes` has the best match with the input phonetic sequence ($\Pr(phs_i|w_i(j)) = 4.92 \times 10^{-4}$). However, owing to the frequencies of the candidates, it ends up in position 2 with probability $\Pr(w_i(j)|phs_i) = 0.264$, while `house` and `hands` obtain higher probabilities. If we use a threshold $T = 0.75$, the top two candidates are retained for evaluation by the language model, while with a threshold $T = 0.5$, the top three candidates are kept.

3.3 Adjustment of the Final Word Hypothesis

The purpose of the search algorithm is to generate sentences for evaluation by the sentence model. Since the complete set of possible sentences is very large, and the search algorithm generates only a small subset of these sentences, it is possible that the correct sentence for a phonetic sequence won't be in that subset (and therefore won't be evaluated by the sentence model). This situation may be exacerbated by the expediency measures we adopted, i.e., the correct words may have been removed by the search algorithm when the short list was created or when it was pruned (Sections 3.1 and 3.2 respectively). However, if we can identify which parts of the final sentence are likely to be incorrect, we could apply a post-processing stage to correct mistakes made during recognition.

Here we propose an adjustment process where the final, top-ranked sentence hypothesis produced by the search system is further analyzed (in principle, it is possible to extend this analysis to lower-ranked sentences, but owing to processing limitations, we restricted ourselves to the top-ranked sentence). Since such an analysis is applied only to a small portion of a sentence, it can be more intensive than the analysis employed during the search, without significantly degrading processing speed.

This adjustment process has three stages: (1) *region identification*, (2) *deep analysis and sentence adjustment*, and (3) *sentence ranking*.

Region identification. This stage identifies a portion of the final sentence hypothesis whose accuracy could significantly increase as a result of in-depth analysis. First, we identify the *nucleus* of the sentence — a word that would most benefit from such an analysis. We then create a larger *phonetic symbol region* that includes the phonetic symbols in that nucleus word — this is the region to be analyzed further (there is little benefit from analyzing the phonetic symbols corresponding just to a single word).

To determine the position and range of a phonetic symbol region in a candidate sentence, we require an estimate of the accuracy of different parts of the sentence. Clearly, we cannot use the word error rate (WER) measure for this purpose, because this would require knowledge of the correct words in the sentence. However, we can indirectly assess WER through the probability of a sentence. Figure 2 shows a graph of WER against the negative log of the probability of a sentence per phoneme for each of the 192 sentences in the TIMIT test set. This metric is calculated as follows.

$$\frac{-\log_2(\Pr(\text{candidate sentence}))}{\#\text{ of phonetic symbols in the sentence}}$$

The graph shows that as the negative log probability (per phoneme) of a sentence increases (its probability decreases), the WER for that sentence tends to increase (the line fitted to this graph has a correlation coefficient of 0.47; the 95% confidence interval for this correlation is $[0.354, 0.575]$). We take advantage of this observation to estimate the accuracy of a found sentence: a high negative log probability per phoneme suggests a high WER. This relationship is also observed when the negative log measure is calculated for a single word or a region of words. When this measure was calculated for each word in the final sentence hypothesis for the 192 TIMIT test sentences, the word with the highest negative log probability per phoneme in a sentence was incorrect 48%

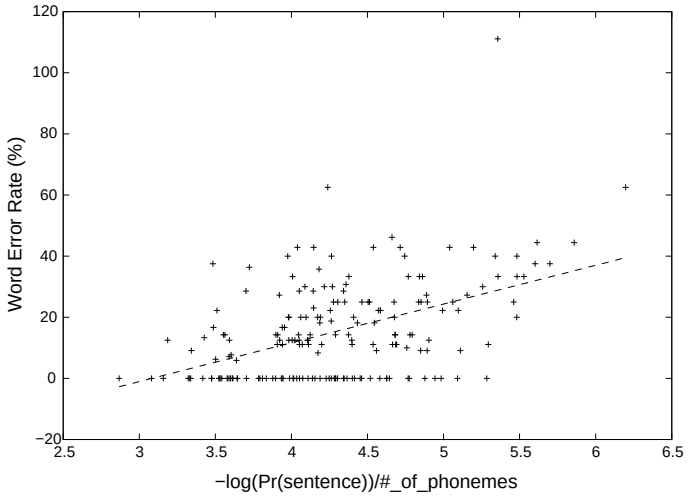

Figure 2: WER versus negative log of sentence probability per phoneme for 192 test sentences

of the time. As shown in Section 4, the average WER for a sentence is substantially lower than 48%.

This result suggests that the word with the highest negative log probability per phoneme is a good starting point to perform a more thorough analysis, hopefully leading to improvements in sentence accuracy. This word is designated as the *nucleus*.[5] The phonetic symbol region is then expanded by one word before the nucleus (unless the nucleus is the first word in the sentence) and one word after the nucleus (unless the nucleus is the last word in the sentence). In order to produce a sufficiently large region for manipulation in the next step, we include an additional word before (or after) the nucleus if the number of phonetic symbols in the word preceding (or following) the nucleus is less than the median number of phonetic symbols in the words in the training text.

Deep analysis and sentence adjustment. This stage performs a rigorous analysis of the phonetic symbols in the chosen region, generating one or more new sentence hypotheses that have new word hypotheses for the phonetic symbols in the analyzed region. The analysis consists of postulating words for this region without applying the efficiency measures used in the initial search procedure (short list and thresholding, Sections 3.1 and 3.2 respectively). The idea is that errors due to this limited search may be corrected by inspecting all the candidates.

The level-building search algorithm described at the beginning of Section 3 is restarted on the phonetic symbol sequence for the chosen region, using as candidates the words in the complete lexicon, rather than a shortlist. The best hypothesis pro-

[5] Another way to implement this idea involves proceeding with the detailed analysis only if the highest negative-log-probability-per-phoneme for a word is significantly larger than the average negative-log-probability-per-phoneme for the entire sentence.

duced by this search is then spliced into the original sentence hypothesis. In order to complete this step, we must take into account the impact of the words in the replaced region on the probability of the sentence according to the language model. To this effect, we re-calculate the conditional probabilities of the words at the left boundary of the new region in the context of existing previous words, and the probabilities of the words following the new region in the context of the words at its right boundary.

Sentence ranking. The most probable sentence hypothesis produced in the previous stage is then ranked within the previous list of sentence hypotheses, and will be selected if its probability is higher than that of the best sentence before adjustment. Although this adjustment often improves the accuracy of the final sentence hypothesis, it is not guaranteed to do so. That is, the adjustment sometimes produces a new sentence whose probability is higher than that of the previous best sentence, but in fact has a higher WER. Nonetheless, overall the adjustment step has a beneficial effect on performance (Section 4).

4 Results

To evaluate the effect of the various operational parameters on system performance, we conducted three sets of experiments as follows.

- **Experiment Set 1** to find the settings for the following factors that achieve the lowest WER.
  - Method for estimating γ for the language model (Equation 9, Section 2.1): the Uniform prior yields $\gamma = 0.5$, and the Dirichlet prior $\gamma = N(w|PoS_i, w_{i-1}) + 0.5$, where $N(w|PoS_i, w_{i-1})$ is the number of different words w that have PoS_i and follow w_{i-1} (Appendix A).
  - Method for estimating $\Pr(phs_i|w_i)$ in the word model: *Weighted* or *Max* (Section 2.2).
  - Method for pruning the word hypothesis list during search: using a dynamic threshold (Section 3.2) versus choosing a fixed number of words. We compared a dynamic threshold of 0.5 with top-two word candidates.
  - Performing a final word-hypothesis adjustment (Section 3.3) versus performing no adjustment.

  A fractional factorial design [3] was used, yielding 8 (2^3) runs to examine the 2^4 options (each factor has two alternatives).
- **Experiment Set 2** to check the effect of each factor in isolation in the context of the optimal settings of the other factors found in the first set of experiments.
- **Experiment Set 3** to check the effect of
  - different values for α and β in Equations 7 and 8 respectively (Section 2.1),

Table 7: Results of Experiment Set 1 and 2

Run	WER (%)	γ Prior Dir/Uni	Pr($phs_i|w_i$) Wei/Max	Pruning Threshold Dyn/Fix	Adjustment Y/N
1	28.4936	Uni	Max	Fix	N
2	19.5182	Dir	Max	Fix	Y
3	24.7945	Uni	Max	Dyn	Y
4	20.6377	Dir	Max	Dyn	N
5	24.7499	Uni	Wei	Fix	Y
6	21.2063	Dir	Wei	Fix	N
7	28.2478	Uni	Wei	Dyn	N
8	**17.4034**	**Dir**	**Wei**	**Dyn**	**Y**
9	25.6185	Uni	Wei	Dyn	Y
10	17.9268	Dir	Max	Dyn	Y
11	18.2909	Dir	Wei	Dyn	N
12	18.0384	Dir	Wei	Fix	Y

- different dynamic thresholds for pruning the word hypothesis list (Section 3.2), and
- using the entire lexicon when postulating candidate words instead of using the short list (Section 3.1).

Table 7 shows the average WER obtained for the 192 TIMIT test sentences for the first two sets of experiments, where WER for a sentence is calculated as follows.

$$\text{WER} = \frac{\text{\# word of insertions, deletions and substitutions}}{\text{\# of words in the sentence}}$$

The first eight rows show the results of the fractional factorial design for Experiment Set 1, and the last four rows show the results of the isolated on/off alterations performed in Experiment Set 2 for the optimal settings obtained in Experiment Set 1. The first column indicates the run number, the second column shows the average WER, Column 3 contains the method for estimating the γ prior distribution (**Dir**ichlet or **Uni**form), Column 4 shows the method used for estimating Pr($phs_i|w_i$) (**Wei**ghted or **Max**), the method used for pruning the word hypothesis list (**Dyn**amic or **Fix**ed) appears in Column 5, and the last column indicates whether a final word-hypothesis adjustment was performed (**Y**es or **N**o).

The first eight runs indicate that the best results are obtained when we use the Dirichlet prior and the Weighted method, employ a dynamic threshold for pruning the candidate word list, and perform adjustments to the final word hypothesis (Run 8). To confirm these insights, we fitted a linear model to the WER obtained from the first eight runs and the additional four runs. These four runs increase the accuracy of the linear model, as they provide focused information about the contribution of each factor in light of the optimal values of the other factors. The resultant linear model had an R-squared value of 0.9673, which means that it describes over 96.7% of the variation

Table 8: Results of Experiment Set 3

Run	WER (%)	Short List	α	β	Dynamic Threshold
13	16.5297	N	0.5	1.0	0.5
14	19.0461	Y	1.0	1.0	0.5
15	19.6925	Y	0.5	0.5	0.5
16	17.0550	Y	0.5	1.0	0.25
17	18.2909	Y	0.5	1.0	0.75

in the data. The model also showed that the WER was significantly decreased when a Dirichlet prior was used instead of a Uniform prior (p-value = 2.56e-06), and when a post-processing adjustment was performed (p-value = 2.63e-03). However, there was no significant decrease in WER when the Weighted method was used instead of the Max method, or when dynamic pruning was performed instead of choosing the top-two candidates.

Using the linear model, we conducted Experiment Set 3 to investigate the effect of other factors on the system's performance (Table 8). Firstly, we investigated whether there was any significant decrease in WER if we used the full lexicon instead of the short list (Run 13). As expected, there was a significant decrease in WER (p-value = 0.02413), but also a significant increase in run time. When using short lists, the word candidate list had an average of 239 words (median = 186, lower quartile = 128, upper quartile = 319), whereas for the full lexicon, the candidate list was always 6,224 words.[6] The total time to interpret the 192 sentences was over double the maximum time taken by the 12 initial runs (which used the shortlist). These times are for the complete interpretation process, of which the comparison and ranking of word hypotheses is one step.[7]

In Run 14, $\alpha = \beta = 1$, and in Run 15, $\alpha = \beta = 0.5$. In both cases we found that there was a significant increase in WER compared to $\alpha = 0.5$ and $\beta = 1$, which were used for the first twelve runs (p-value = 4.6e-4 and p-value = 6.86e-5, respectively). Recall that α is the flattening constant for triples $(w_i, \mathrm{PoS}_i, w_{i-1})$, and β for tuples (w_i, PoS_i). These findings indicate that higher flattening constants are suitable for larger groupings (which have more occurrences of the values in question). Finally, we considered the effect of changing the value of the dynamic pruning threshold from 0.5 (which was used in Experiment Sets 1 and 2) to 0.25 (Run 16) and to 0.75 (Run 17). We found that there was a decrease in WER when we decreased the threshold (thereby including more words in the candidate list), but this reduction was not significant (p-value = 0.23999), and it increased processing time by 50% on average. On the other hand, there was a significant increase in WER (p-value = 8.449e-03) when we increased the threshold.

[6] Note that the average of 239 words differs from the class size average of 496 reported in Section 3.1.1. This is because here we consider the number of words actually compared with input phonetic sequences, which in practice were obtained from the smaller classes.

[7] The actual time to interpret a sentence is quite long, as our system uses legacy code which runs on machines that are quite old. However, the principles underlying the design are independent of these specifics.

5 Related Work

Although phoneme recognition and speaker recognition studies have been performed using the TIMIT corpus, there are only a few studies using the TIMIT corpus that are directly comparable to our own. Wang [52] presented a simple word-pair (bigram) grammar and a beam search algorithm as part of a speech recognition system that utilizes prosodic information. Performing this search (without any acoustic or pronunciation modeling), he achieved an accuracy of between 79.07% and 87.86% on TIMIT's standard test set (the accuracy increases as the size of the beam increases — as does the processing time). Unlike our system, this experiment did not model the possible pronunciation of words as phonetic sequences. Ristad and Yianilos [40] argued for a pronunciation model comprising small variations to pronunciations found in training data. They achieved a 17.36% word error rate on TIMIT's standard test set, but unlike our system, they had the advantage of knowing the correct segmentation of the input phonetic sequence into words. Furthermore, they analyzed isolated words rather than complete sentences, and hence did not use a language model. We achieve results that are similar to those presented for these systems, even though we address a less constrained problem.

Like our system, these systems perform a subset of the standard subtasks in speech recognition. Other systems use the TIMIT corpus more extensively, training models in the low acoustic level through to the high sentence level. The use of acoustic and phonetic subsystems provides a measure for confidence in the identity of the phonetic symbols that constitute the input to higher level tasks, which in turn may improve overall results. Wang and Pols [53] incorporated duration features for phones and used specific word junction models to achieve a 79.90% word recognition rate on TIMIT's standard test set. Antoniou [1] describes a modular neural network architecture that comprises individual neural networks for each phone and uses seven broad phone classes to capture acoustic contextual information. This system achieved a word recognition accuracy of 77.0% on the TIMIT core test set. The system described in [25] supplements a fixed set of pronunciation variants for each word with extra variants added dynamically during the recognition of utterances. This hybrid system, which uses a HMM for acoustic modeling, a neural network for phonetic modeling, and bigrams for the language model, achieved a word error rate of 22.0% on TIMIT's core test.

More recently, higher accuracies were obtained by Yang *et al.* [56], Grebenskaya *et al.* [18], and Mporas *et al.* [30]. Building on results reported in [6] and [55], Yang *et al.* [56] describe a method for constructing a lexicon of pronunciation variants by applying probabilistic rewrite rules to basic forms. These rules are learned from training utterances. They use HMMs for acoustic models and apply a backing-off bigram model for language, achieving a WER of 6.10% for the phonetically compact sentences in the TIMIT core test set, and 7.03% for the phonetically diverse sentences. Grebenskaya *et al.* [18] achieved a WER of 6.18% for the TIMIT core test sentences by augmenting HMM-based speech recognition models with speaker classification obtained by clustering vector quantization codebooks. Finally, Mporas *et al.* [30] achieved a WER of 7.88% on TIMIT's standard test set, using the open-source Sphinx-4 speech

recognizer[8] and the CMU-Cambridge Statistical Modeling Toolkit.[9] These WERs are significantly better than those obtained by our system. However, the main contribution of our work is not in the overall lexical access accuracy we obtained, but in the insights regarding the influence of various modeling and search parameters on performance, which are applicable to statistical lexical access systems.

6 Conclusions

The difficulties of the lexical access problem stem from the inherent complexity and variability of real speech: the input speech signal is subject to mis-recognitions and mispronunciation. This causes the signal to vary markedly from its canonical form. Furthermore, phonetic symbols, syllables, words and the boundaries of words may be slurred or omitted due to co-articulation. In this paper, we have considered a statistical approach to the lexical access problem, focusing on the issues pertaining to the training of statistical models, and on the difficulties associated with the large search space. We have offered an experimental study that compares the effect of two sentence modeling parameters and three search heuristics on lexical access performance. Specifically, the sentence modeling parameters pertain to (1) the estimation of the back-off parameter in the deleted interpolation model (Section 2.1), and (2) the calculation of the probability of a particular word realization during word modeling (Section 2.2). The heuristics for reducing the number of word candidates consist of (1) generating a short-list of likely possibilities (Section 3.1), and (2) pruning this list after it has been evaluated by the word model (Section 3.2). In addition, we investigated the impact of performing regional word adjustments during a post-processing step (Section 3.3).

Our studies have yielded insights that are applicable to other statistical lexical access systems. Specifically, our results show that the best performance was obtained when we use the Dirichlet prior for estimating the back-off factor during language modeling, and the Weighted method for calculating the probability of a particular word during word modeling; when we employ a dynamic threshold for pruning the candidate word list, and perform adjustments to the final word hypothesis. These effects were statistically significant for the Dirichlet prior and the post-processing adjustment, but not for the other two parameters. Additionally, as expected, further improvements in processing time (at the expense of accuracy) were obtained using a short-list of word candidates, while improvements in accuracy (at the expense of processing time) were obtained with more lenient dynamic thresholds.

Acknowledgements

This research was supported in part by grant N016/099 from the Australian Telecommunications and Electronics Research Board, and ARC Fellowship F39340111. The authors thank Jonathan Oliver for his assistance with early versions of this work.

[8] `http://cmusphinx.sourceforge.net/sphinx4/`
[9] `http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html`

References

[1] C. Antoniou. Modular neural networks exploit large acoustic context through broad-class posteriors for continuous speech recognition. In *ICASSP-01 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 505–508, Salt Lake City, Utah, 2001. IEEE.

[2] T.C. Bell, J.G Cleary, and I.H. Witten. *Text Compression*. Prentice Hall, 1990.

[3] George E. P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, 2005.

[4] N. Carver and V. Lesser. The evolution of blackboard control architectures. *Expert Systems with Applications – Special Issue on the Blackboard Paradigm and Its Applications*, 7(1):1–30, 1994.

[5] F. Chen. Lexical access and verification in a broad phonetic approach to continuous digit recognition. In *ICASSP-86 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1089–1092, Tokyo, Japan, 1986. IEEE.

[6] N. Cremelie and J. Martens. In search of better pronunciation models for speech recognition. *Speech Communication*, 29:115–136, 1999.

[7] Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on Information Theory*, 48:1518–1569, Jun 2002.

[8] M.E. Epstein. *Statistical Source Channel Models for Natural Language Understanding*. PhD thesis, New York University, 1996.

[9] U. Essen and V. Steinbiss. Co-occurrence smoothing for stochastic language modeling. In *ICASSP-92 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 161–164, San Fransisco, California, 1992. IEEE.

[10] W.M. Fisher, M. Doddington, and K.M. Goudie-Marshell. The DARPA speech recognition database: Specifications and status. In *Proceedings of the DARPA Speech Recognition Workshop*, Report No. SAIC-86/1546, pages 93–99, 1986.

[11] L. Fissore, P. Laface, G. Micca, and R. Pieracci. Lexical access to large vocabularies for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(8):1197–1213, 1989.

[12] L. Fissore, G. Micca, and R. Pieraccini. Strategies for lexical access to very large vocabularies. *Speech Communication*, 7:355–366, 1988.

[13] A. Gallardo-Antolín, J. Ferreiros, J. Macías-Guarasa, R. De Córdoba, and J. M. Pardo. Incorporating multiple-HMM acoustic modeling in a modular large vocabulary speech recognition system in telephone environment. In *ICSLP-00 – Proceedings of the Sixth International Conference on Spoken Language Processing*, pages 827–830, Beijing, China, 2000.

[14] J.S. Garofolo, L.F. Lamel, W.N. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren. DARPA TIMIT – acoustic-phonetic continuous speech corpus. Technical Report NISTIR 4930, US Department of Commerce, National Institute of Standards and Technology, 1993.

[15] J.L. Gauvain, L.F. Lamel, G. Adda, and M. Adda-Decker. Developments in continuous speech dictation using the ARPA WSJ task. In *ICASSP-95 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 65–68, Detroit, Michigan, 1995.

[16] Irving John Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. Research Monograph No. 30. MIT Press, Cambridge, Massachusetts, 1965.

[17] D.B. Grayden and M.S. Scordilis. Phonemic segmentation of fluent speech. In *ICASSP-94 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 73–76, Adelaide, Australia, 1994.

[18] O. Grebenskaya, T. Kinnunen, and P. Franti. Speaker clustering in speech recognition. In *FINSIG'05 – Proceedings of the 2005 Finnish Signal Processing Symposium*, pages 46–49, Kuopio, Finland, 2005.

[19] D.P. Huttenlocher and V.W. Zue. A model for lexical access from partial phonetic information. In *ICASSP-84 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 391–394, San Diego, California, 1984. IEEE.

[20] P. Jeanrenaud, E. Eide, U. Chaudhari, J. McDonough, K. Ng, M. Siu, and H. Gish. Reducing word error on conversational speech from the switchboard corpus. In *ICASSP-95 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 53–56, Detroit, Michigan, 1995.

[21] F. Jelinek and R.L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Publishing Company, 1980.

[22] T. Kaneko and N.R. Dixon. A hierarchical decision approach to large-vocabulary discrete utterance recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31:1061–1066, 1983.

[23] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 400–401, March 1987.

[24] T. Laureys, K. Demuynck, J. Duchateau, and P. Wambacq. An improved algorithm for the automatic segmentation of speech. In *LREC-02 – Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1564–1567, Las Palmas, Canary Islands, 2002.

[25] K. Lee and C. Wellekens. Dynamic lexicon using phonetic features. In *Proceedings of Eurospeech-01*, pages 1413–1416, Aalborg, Denmark, 2001.

[26] S. C. Levinson. *Mathematical Models for Speech Technology*. Wiley, 2005.

[27] T.W. Lewis and D.M.W. Powers. Distinctive feature fusion for improved audio-visual phoneme recognition. In *ISSPA 2005 – Proceedings of the IEEE 8th International Symposium on Signal Processing and its Applications*, pages 62–65, Sydney, Australia, 2005.

[28] W.D. Marslen-Wilson and A. Welsh. Processing interactions and lexical access during word recognition in continuous speech. *Cognitive Psychology*, 10:29–63, 1978.

[29] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[30] I. Mporas, T. Ganchev, E. Kotinas, and N. Fakotakis. Examining the influence of speech frame size and number of cepstral coefficients on the speech recognition performance. In *SPECOM'2007 – Proceedings of the Speech and Computer Conference*, pages 134–139, Moscow, Russia, 2007.

[31] H. Murveit, M. Weintraub, M. Cohen, J. Bernstein, and A. Rudnicky. Lexical access with lattice input. In *ICASSP-87 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 837–840, Dallas, Texas, 1987.

[32] C.S. Myers and L.R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 2, pages 284–297. IEEE, April 1981.

[33] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino-acid sequence of two proteins. *Journal of Molecular Biology*, (48):443–453, 1970.

[34] H. Ney and U. Essen. On smoothing techniques for bigram-based natural language processing. In *ICASSP-91 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 825–828, Toronto, Canada, 1991. IEEE.

[35] D. Norris. Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52:189–234, 1994.

[36] F. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *ACL-02 – Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, 2002.

[37] D.B. Pisoni, H.C. Nusbaum, P.A. Luce, and L.M. Slowiaczek. Speech perception, word recognition and the structure of the lexicon. *Speech Communication*, 4:75–95, 1985.

[38] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, N.J, 1993.

[39] L.R. Rabiner, B.H. Juang, and C.H. Lee. An overview of automatic speech recognition. In C.H. Lee, F.K. Soong, and K.K. Paliwal, editors, *Automatic Speech and Speaker Recognition: Advanced Topics*, pages 1–30. Kluwer Academic Publishers, 1996.

[40] E.S. Ristad and P.N. Yianilos. A surficial pronunciation model. In *Proceedings of the ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pages 117–120, Kerkrade, The Netherlands, 1998.

[41] A.I. Rudnicky. An unanchored matching algorithm for lexical access. In *ICASSP-88 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 469–472, New York, New York, 1988. IEEE.

[42] D. Sankoff and J.B. Kruskal. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison Wesley, London, 1983.

[43] S. Seneff. The use of linguistic hierarchies in speech understanding. In *ICSLP-98 – Proceedings of the Fifth International Conference on Spoken Language Processing*, page 12, Sydney, Australia, 1998.

[44] D.W. Shipman and V.W. Zue. Properties of large lexicons: Implications for advanced isolated word recognition systems. In *ICASSP-82 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 546–549, Paris, France, 1982. IEEE.

[45] M. Tang, S. Seneff, and V.W. Zue. Modeling linguistic features in speech recognition. In *Proceedings of Eurospeech-03*, pages 2585–2588, Geneva, Switzerland, 2003.

[46] I.E. Thomas. *An Information-Theoretic Approach to Speech Recognition*. PhD thesis, Monash University, Clayton, Victoria, Australia, 2004.

[47] I.E. Thomas, I. Zukerman, and B. Raskutti. Extracting phoneme pronunciation information from corpora. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pages 175–184, Sydney, Australia, 1998.

[48] V.T. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4(1):52–57, 1968.

[49] A. Waibel. Suprasegmentals in very large vocabulary isolated word recognition. In *ICASSP-84 – Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 387–390, San Diego, California, 1984. IEEE.

[50] C.S. Wallace. *Statistical and inductive inference by Minimum Message Length*. Springer, 2005.

[51] C.S. Wallace and D.M. Boulton. An information measure for classification. *Computer Journal*, 11:185–194, 1968.

[52] X. Wang. *Integrating Knowledge on Segmental Duration in HMM-based Continuous Speech Recognition*. PhD thesis, University of Amsterdam, 1997.

[53] X. Wang and L. Pols. Word juncture modeling based on the TIMIT database. In *Proceedings of Eurospeech-97*, pages 2407–2410, Rhodes, Greece, 1997.

[54] I.H. Witten and T.C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. In *IEEE Transactions on Information Theory*, volume 37:4, pages 1085–1094, July 1991.

[55] Q. Yang and J.P. Martens. On the importance of exception and cross-word rules for the data-driven creation of lexica. In *Proceedings of the 11th IEEE ProRisc Workshop on Automatic Speech Recognition*, pages 589–593, Veldhoven, The Netherlands, 2000.

[56] Q. Yang, J.P. Martens, P.J Ghesquiere, and D. Van Compernolle. Pronunciation variation modeling for ASR: Large improvements are possible but small ones are likely to achieve. In *ISCA Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*, pages 123–128, Estes Park, Colorado, 2002.

Appendix A: Estimates for the Word Model

The distribution of the words w_i together with the escape codes in Section 2.1 can be represented by a multinomial distribution. Let $\Pr(w_i)$ be the probability of observing word w_i, $i = 1,\ldots,m$, and $\Pr(w_{m+1})$ be the probability of observing the escape code. Then the multinomial distribution is:

$$\frac{(n_1+\cdots+n_{m+1})!}{n_1!\cdots n_{m+1}!}\Pr(w_1)^{n_1}\cdots\Pr(w_{m+1})^{n_{m+1}}$$

where $n_1,\ldots n_{m+1}$ are the frequencies of the words and the escape code. Now, a suitable prior for the multinomial distribution is its conjugate prior, viz the Dirichlet distribution:

$$\frac{\Gamma(\omega_1+\cdots+\omega_{m+1})}{\Gamma(\omega_1)\cdots\Gamma(\omega_{m+1})}\Pr(w_1)^{\omega_1-1}\cdots\Pr(w_{m+1})^{\omega_{m+1}-1}$$

where Γ is the Gamma function, and $\omega_1,\ldots,\omega_{m+1}$ are parameters.

Using the Minimum Message Length criterion [50] and the conjugate prior, we obtain the following estimates for $\Pr(w_1),\ldots,\Pr(w_{m+1})$.

$$\widehat{\Pr}(w_i) = \frac{n_i+\omega_i-0.5}{\sum_{i=1}^{m+1}(n_i+\omega_i-0.5)} \qquad i = 1,\ldots,m+1$$

Note that these estimates depend upon the parameters $\omega_1,\ldots,\omega_{m+1}$ chosen to describe the prior. Moreover, it can then be seen that the different values of α, β and γ in

Equations 7–9 correspond to different values of $\omega_1, \ldots, \omega_{m+1}$ and consequently different priors. In particular, the uniform prior corresponds to the parameter values $\omega_1 = \cdots = \omega_{m+1} = 1$. While the estimates corresponding to the Dirichlet prior (referred to as simply Dirichlet throughout the paper) correspond to the parameter values $\omega_1 = \cdots = \omega_m = 1$ and $\omega_{m+1} = m + 0.5$.