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 ones 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

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.

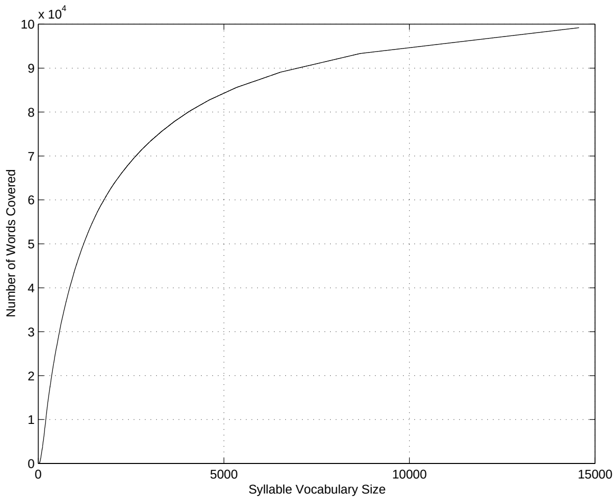

Figure 1: Word coverage versus syllable vocabulary size.

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.