KEYWORD SPOTTING OF ARBITRARY WORDS USING MINIMAL SPEECH RESOURCES

Alvin Garcia and Herbert Gish

BBN Technologies
10 Moulton Street
Cambridge, MA 02138, USA
{agarcia,hgish}@bbn.com

ABSTRACT

Traditional approaches to keyword spotting employ a large vocabulary speech recognizer, phone recognizer or a whole-word approach such as whole-word Hidden Markov Models. In any of these approaches, considerable speech resources are required to create a word spotting system. In this paper we describe a keyword spotting system that requires about fifteen minutes of word-level transcriptions of speech as its sole annotated resource. The system uses our self-organizing speech recognizer that defines its own sound units as a recognizer for the speech in the speech domain under consideration. The transcriptions are used to train a grapheme-to-sound-unit converter. We describe this novel system and give its keyword spotting performance.

1. INTRODUCTION

As the amount of human communications stored as recorded speech continues to increase in a multiplicity of languages and acoustic domains (e.g. "pod-casts", radio and television news archives), efficient means for finding speech of interest are increasingly important. The basic approach to locating speech passages of interest is through the occurrence of relevant keywords. Traditional approaches to keyword spotting are usually based on large vocabulary word recognizers, phone recognizers, or whole-word models that either use HMMs or word templates. In each of these approaches a significant amount of resources is required for building a useful word spotter. Word recognizers require tens of hours of word-level transcriptions as well as a pronunciation dictionary. Phone recognizers require phone marked transcriptions and whole-word recognizers require word markings for each of the keywords. All of these types of annotations can be rather time consuming. Also, word recognizers are, of course, unable to find words that are out-of-vocabulary (OOV). OOV keywords are a common problem in many applications, as keywords of interest can be quite transitory (e.g. Google's weekly Zeitgeist [1]). To handle OOV keywords, spotting approaches based upon phone-level recognition have been applied to supplement or replace systems based upon large vocabulary recognition [2].

For many applications of interest, such resources are not readily available, if at all, rendering these traditional approaches unusable. The most obvious case is when we are dealing with a language for which the resources have never been developed, but it also includes the situation where the resources that are available are acoustically mismatched to the domain of interest. Such mismatches can result in severe performance degradations. This can occur when the collection to be searched contains speech in channel conditions other than the telephony or television audio channels common for these corpora.

We present a novel approach capable of operating in such extremely-limited-resource scenarios. All that is required is the collection of speech to be searched and word-level transcriptions for a very limited amount, as little as 15 minutes worth, of this data.

The system that we develop is based on segmental models [3]. The segmental model allows us to develop a phone-like speech recognizer, automatically and without supervision, on a speech corpus. The essential steps, of which more will be said below, are segmentation, clustering and finally, using the clusters, the creation of a segmental Gaussian mixture model (SGMM). A SGMM is a Gaussian mixture model where the mean of each mixture term is a time-normalized quadratic trajectory in feature space that represents a sound unit of the language. These units we term "discovered-units" and they are phone-like and syllable-like units. This segmental mixture model is used as a decoder where segmented input speech is assigned the index of the maximum term of the mixture [3]. Thus the decoded speech is in terms of the SGMM indices.

We use the resulting segmental mixture model to decode the speech recordings to be searched, generating transcriptions in terms of the discovered-units. We also decode a limited subset of speech recordings, for which parallel text transcriptions are available. Given these parallel transcriptions, the Joint Multigram Model [4] is used to obtain a probabilistic mapping between sequences of letters and sequences of discovered-units. This model is then used to predict the "pronunciation" of a given keyword, in terms of the discovered-units modeled by the segmental model, thereby eliminating the need for a pronunciation dictionary. Finally, a dynamic-programming search, which minimizes the string edit distance between the predicted pronunciation of the keyword, and the discovered-unit transcription of each speech recording in the collection, is used to find putative occurrences of the keyword.

Subsequent sections of this paper describe the various components of the new approach and present experimental results.

This work was funded by ARDA, under the "Speech Processing for Question Answering" contract, NBCHC040138.

2. SYSTEM COMPONENTS

This section briefly describes the various components of our system for word-spotting with limited speech resources.

2.1. Subword-Unit Modeling of Speech via the Segmental Speech Model

The major system components of the segmental speech model are:

1) Segmenter: The segmenter segments all training and test data based on the the occurence of spectral discontinuities. The segments are nonoverlapping and variable duration. It produces segments that are phone-like and syllable-like. It has no user-settable parameters and learns how to segment based on statistcal models that are estimated from the acoustics of the speech domain to which it is applied. It is related to the type of model originally developed by J. Cohen [5].

2) Clustering Algorithm: The clustering algorithm clusters segments into a prescribed number of clusters equal to the number of sound units that will be used to represent the data. Each segment is modeled by a polynomial trajectory/segment model and the distance between segment models defines the distance between segments for the clustering algorithm. Specifically, a quadratic polynomial is used to model the time-varying trajectory of the cepstral features. That is, for a segment of length N frames, each feature dimension i is modeled as:

$$c_i(n) = \mu_i(n) + e_i(n), \qquad n = 1 \ldots N \tag{1}$$

where $c_i(n)$ are the observed cepstral coefficients for the ith feature dimension, and $\mu_i(n)$ is the quadratic polynomial:

$$\mu_i(n) = b_{i1} + b_{i2}n + b_{i3}n^2, \qquad n = 1 \ldots N \tag{2}$$

Since we deal with varialble length segments the actual modeling equations are modified to normalize all equations to unit length. The details are contained in [3].

3) Segmental Gaussian Mixture Model (SGMM): As previously noted the SGMM [6] works as the speech recognizer. Each cluster that results from the clustering algorithm initializes a term of the SGMM and it is trained by the EM algorithm.

The details regarding estimation of parameters of the segmental model are deferred to the above reference. Of particular interest here is the fact that the creation of the SGMM does not require any transcriptions or phonetic labeling for training. It is a completely self-organizing, unsupervised process. Because of this, the SGMM can be trained on speech from the collection of speech recordings to be searched, thereby eliminating any acoustic mismatch between training and operation conditions. As stated earlier, in the case of a traditional large vocabulary or phone recognizer approach, there will be a degradation in recognition accuracy, and subsequently keyword spotting performance, if the collection of speech recordings to be searched is acoustically mismatched from the corpora used to train the recognizer.

In the system we describe here, we use this SGMM as a simple decoder, which labels each input segment with the index of the most likely model component, i.e. discovered-unit. We use a fully ergodic recognition network, in which any discovered-unit can follow any other one, and a null grammar. The resultant sequence of discovered-unit indices serves as a subword-unit transcription of the corresponding speech in terms of the subword units found in the training speech.

2.2. Grapheme-to-Discovered-Unit Mapping via the Joint Multigram Model

As stated earlier, for some low-resource languages, no pronunciation dictionary may be available. Without such a dictionary, a traditional large vocabulary speech recognizer cannot be trained. In our approach, we do away with the requirement of a pronunciation dictionary by building a grapheme-to-"phoneme" model, which we then use to predict the pronunciation of a desired keyword, in terms of the discovered-units modeled by our segmental model.

In particular, we employ the *Joint Multigram Model* described in [4]. The joint multigram model is a statistical model for mapping between variable-length symbol sequences in one stream O and variable-length symbol sequences in another stream Ω. A joint multigram model with parameters n and ν establishes a mapping between symbol sequences of length $1 \ldots n$ in O to symbol sequences of length $1 \ldots \nu$ in Ω. The model is estimated using an iterative maximum-likelihood estimation procedure, and the resultant model can then be used to map sequences in the alphabet of symbol stream O into a corresponding sequence in the alphabet of the other symbol stream Ω.

In our case, we use the joint multigram model to model a correspondence between sequences of letters in the word-level text transcriptions of some training speech, and sequences of discovered-units in the segmental model transcriptions of the same speech. Then, given an arbitrary keyword, its sequence of letters is mapped into a sequence of discovered-units corresponding to its most likely pronunciation. Thus, not only is no pronunciation dictionary necessary, but the keyword need not even have been observed in the training data. Since there is no pre-defined dictionary, there are no OOV keywords.

2.3. Dynamic Programming Search

Given a keyword pronunciation predicted by the joint-multigram model above, we search for its sequence of discovered-units in the discovered-unit transcriptions of the speech recordings to be searched. In that both the predicted pronunciation of the keyword and the discovered-unit transcriptions of the speech recordings may contain errors, we utilize a string-searching algorithm which allows for insertions, deletions, and substitutions in the search. We employ a dynamic programming approach, known as the Smith-Waterman algorithm in the computational biological literature [7], which minimizes the edit distance between two strings. In this case the strings being matched are the the keyword's predicted pronunciation, and the discovered-unit transcription of a given speech recording. The algorithm searches for the best starting point of the keyword pronunciation string in the typically longer discovered-unit transcription of speech recording. The resultant edit distance is used as the score for the given recording, with lower scores indicating better matches.

To mitigate the effects of errors in the predicted pronunciations, we consider the top-N pronunciations predicted by the joint-multigram model, and take the one yielding the minimum edit distance with respect to a given speech recording.

Also, as others have done with phonetic-recognition based approaches to mitigate the effects of decoding errors, we utilize lattice decode results rather than top-1 decoding results. However, unlike traditional phone recognizers, segmentation in our system is performed separately and prior to the actual decoding. Consequently, our decode lattices are of the "sausage-link" network form, with multiple hypotheses spanning the a given segment. The Smith-Waterman search algorithm is extended to consider the cost associated with each possible symbol in a given sequence position, rather than the single symbol in the case of top-1 decode results.

3. EXPERIMENTS

We evaluated our system using data from the Spanish Call-Home corpus [8]. To train the (64 component) SGMM, we used two hours of audio, equally sampled from thirty different speakers from this corpus. We then used the segmental model to decode the corpus of test utterances to be searched, generating a discovered-unit transcription for each utterance.

Our test data consisted of 10,000 utterances, with an average length of 8.1 words each, totaling approximately 7 hours worth of speech. From these 10,000 utterances, we searched for occurrences of six different keywords listed in Table 1, shown along with their frequency of occurrence in the test set and in the 15-minute training set.

keyword	frequency in test	frequency in training
dinero	36	0
familia	15	2
problema	48	1
telefono	64	2
trabajo	54	1
universidad	31	0

Table 1. Keywords and frequency of occurrence in test set and 15-minute training set

3.1. Minimal Training Data

We also used the segmental model to decode a separate collection of utterances to use, in conjunction with the corresponding text transcriptions, to train a joint multigram model. We used 15 minutes of transcribed data to train a $n = 2, \nu = 2$ joint multigram model.

Figures 1 – 6 show the ROC curves obtained for the six query words shown in Table 1. Superimposed on the same graphs is the 45-degree diagonal corresponding to random classification results. As these graphs indicate, detection performance is substantially better than random. It is worth pointing out that for two of the keywords, "dinero" and "universidad", there are no examples of the keyword in the joint multigram model training data, as indicated in Table 1. This demonstrates that a keyword can still be reasonably detected even without it having been present in the training data. This is important since, given the limited amount of training data, many keywords of interest may not have been present in the training data.

One useful way to summarize the detection performance of this system is the following Figure Of Merit (FOM): The

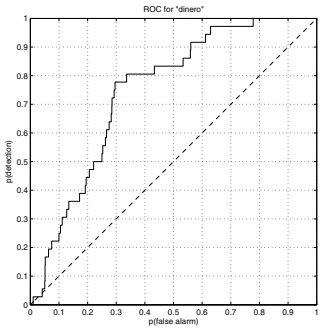

Fig. 1. ROC for "dinero" using 15 min. of transcribed data

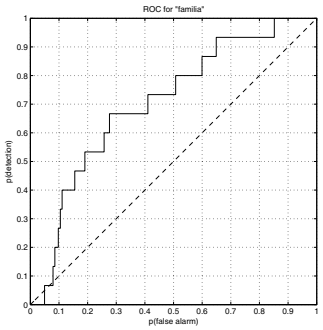

Fig. 2. ROC for "familia" using 15 min. of transcribed data

area below the ROC curve, in the "low" false-alarm rate range ($0 \leq p(\text{false alarm}) \leq 0.3$), normalized by 0.3, the area under the same region of the ROC curve when detection is perfect. The FOM value in the case of random classification is 0.15. Table 2 shows the FOM values for each of the keywords.

3.2. Increased Training Data

The above experiments were repeated using increasing amounts of training data to measure what benefit might be obtained if more transcribed data were available. A $n = 2, \nu = 2$ joint multigram model was trained for each amount of training data. Table 3 shows the individual and average FOM values obtained for 1/2 hour, 1 hour, 2 hours, and 5 hours worth

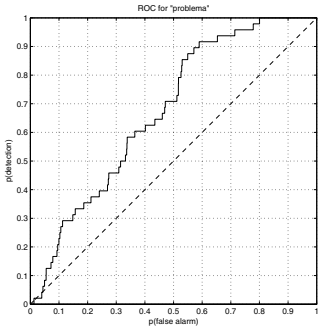

Fig. 3. ROC for "problema" using 15 min. of transcribed data

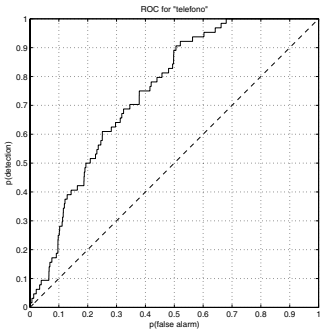

Fig. 4. ROC for "telefono" using 15 min. of transcribed data

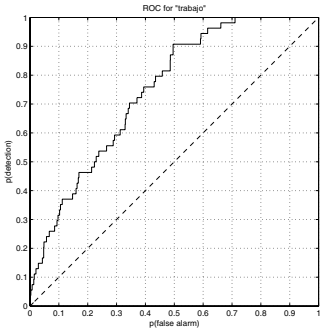

Fig. 5. ROC for "trabajo" using 15 min. of transcribed data

of transcribed training data. As this table shows, the per-keyword FOM can vary substantially as a function of amount of training data. Also, we see that the average FOM value does not increase monotonically as we increased the amount of training data. This result was not expected and suggests that perhaps we are not making the best use of the available amount of training data.

4. CONCLUSIONS

We presented a novel approach for building a keyword spotting system when only a very limited amount of transcribed training data, and no pronunciation dictionary, are available. With such limited resources, traditional keyword spotting approaches, based upon large vocabulary recogizers, phone-recognizers, or whole-word models cannot be applied. Furthermore, since our approach uses no dictionary, there are no OOV keywords: the choice of keywords can be completely arbitrary, and the keyword need not have been present in the limited amount of transcribed training data.

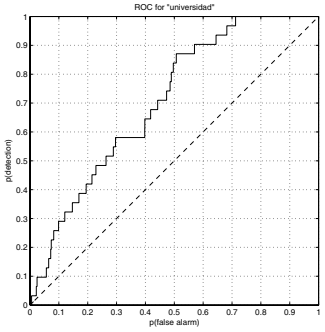

Fig. 6. ROC for "universidad" using 15 min. of transcribed data

keyword	FOM value
dinero	0.34
familia	0.35
problema	0.27
telefono	0.36
trabajo	0.37
universidad	0.32
average	0.34

Table 2. Figure-Of-Merit (FOM) for keywords using 15-minute training set

keyword	0.5 hour	1 hour	2 hours	5 hours
dinero	0.36	0.46	0.29	0.33
familia	0.38	0.35	0.28	0.52
problema	0.27	0.30	0.27	0.29
telefono	0.26	0.30	0.35	0.40
trabajo	0.31	0.39	0.43	0.39
universidad	0.35	0.35	0.26	0.27
average	0.32	0.36	0.32	0.37

Table 3. FOM for keywords using varying amounts of training data

5. REFERENCES

[1] Google.com, "Google press center: Zeitgeist [online]. available: http://www.google.com/press/zeitgeist.html [accessed october 19, 2005]," 2005.

[2] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," *Proc. ICASSP '94*, vol. 1, pp. 377–380, 1994.

[3] Herbert Gish and Kenney Ng, "A segmental speech model with applications to word spotting," *Proc. ICASSP '93*, vol. 2, pp. 447–450, 1993.

[4] Sabine Deligne, Francois Yvon, and Frederic Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," *Proc. EUROSPEECH '95*, pp. 2243–2246, 1995.

[5] Jordan R. Cohen, "Segmenting speech using dynamic programming," *Journal of the Acoustical Society of America*, vol. 69, no. 5, pp. 1430–1438, 1981.

[6] Herbert Gish and Kenney Ng, "Parametric trajectory models for speech recognition," *Proc. ICSLP '96*, vol. 1, pp. 466–469, 1996.

[7] Temple F. Smith and Michael S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[8] Alexandra Canavan and George Zipperlen, *CALLHOME Spanish Speech*, Linguistic Data Consortium, 1981.