A MULTI-CLASS APPROACH FOR MODELLING OUT-OF-VOCABULARY WORDS

Issam Bazzi and James Glass

Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, Massachusetts 02139, USA
{issam, glass}@mit.edu

ABSTRACT

In this paper we present a multi-class extension to our approach for modelling out-of-vocabulary (OOV) words [1]. Instead of augmenting the word search space with a single OOV model, we add several OOV models, one for each class of words. We present two approaches for designing the OOV word classes. The first approach relies on using common part-of-speech tags. The second approach is a data-driven two-step clustering procedure, where the first step uses agglomerative clustering to derive an initial class assignment, while the second step uses iterative clustering to move words from one class to another in order to reduce the model perplexity. We present experiments within the JUPITER weather information domain. Results show that the multi-class model significantly improves performance over using a single OOV class. For an OOV detection rate of 70%, the false alarm rate is reduced from 5.3% for a single class to 2.9% for an eight-class model.

1. INTRODUCTION

Current continuous speech recognition systems are designed to use a vocabulary with a finite set of words. Given a finite vocabulary, the presence of out-of-vocabulary (OOV) words is inevitable. In our JUPITER weather system, for example [3], the word error rate (WER) on data containing OOV words is nearly five times greater than on those containing only in-vocabulary (IV) words. While part of the WER increase is due to poor language modelling of out-of-domain queries, it is clear that OOV words cause recognition errors, and that an ability to identify OOV words would be beneficial. In this research we are exploring a tactic that incorporates an explicit OOV word model into the word-based recognizer [1], where an OOV word can be predicted by a word-level language model. The model is based on a set of subword units capable of generating new phone sequences, most importantly, those outside the vocabulary of the recognizer. In [2], we presented a method to automatically derive a set of variable-length units for the OOV model. We also presented dictionary-based methods for estimating n-grams for use within the OOV network. Both of these efforts produced significant improvements in OOV detection on our weather information task.

In this paper we present a multi-class extension to our approach. Instead of augmenting the word search network with a single OOV model, we add several models, each corresponding to a class of OOV words. We explore two approaches for designing the OOV classes. The first approach relies on using part-of-speech (POS) tags. The second uses a two-step clustering procedure to derive OOV word classes. Both approaches show significant improvement in performance over the single class approach.

The remainder of the paper is organized as follows: we first review the OOV recognition framework. Next we describe how the framework can be extended to model multiple classes of OOV words, and the two approaches for designing OOV word classes. Finally, we present and discuss the results of a series of experiments in the JUPITER domain.

This material is based upon work supported by a graduate fellowship from Microsoft Corporation, and by DARPA under contract N66001-99-1-8904 monitored through NCCOSC.

2. RECOGNITION FRAMEWORK

This section gives a short review of the recognition framework. Details are presented in [1]. To allow for OOV words the recognizer vocabulary is augmented with a generic word model W_{OOV}. This generic word model is considered in parallel with all other words during recognition. The language model of the hybrid recognizer remains word-based, but now includes an entry for W_{OOV}. Since W_{OOV} is part of the vocabulary, the n-gram treats it like any other word in the vocabulary. The FST representation of the recognizer search space in this OOV framework is given by:

$$R_H = C \circ P \circ (L \cup (L_u \circ G_u \circ T_u))^* \circ G_1 \qquad (1)$$

where C represents the mapping from context-dependent to context-independent phonetic units, P represents the phonological rules, and L is the word lexicon. The term $L_u \circ G_u \circ T_u$ represents the OOV model, where L_u is the subword lexicon used to constrain the OOV network to some subword units. G_u is a subword n-gram, and T_u provides hard topological constraints on the model such as imposing a minimum or maximum length requirement. G_1 is the word level n-gram with the single OOV entry. The subscript 1 indicates the *single* OOV class is modelled in the word n-gram. Note that this formulation assumes that all OOV words belong to the same class of words and hence a single model is used to handle all OOV words.

3. THE MULTI-CLASS APPROACH

One of the motivations for extending the framework to model multiple classes of OOV words is to better model the contextual relationship between the OOV word and its neighboring words. Another motivation is to create multiple classes of words such that in each class, words that share the same or similar phone sequences

are grouped together and used to train a class-specific subword n-gram language model. To extend our approach to model multiple classes, we can construct multiple generic word models and create a search network that allows for either going through the IV branch or through any of several OOV branches each representing a class of OOV words. Suppose we have N classes of OOV words that we are interested in modelling. If we construct N subword search networks, one for each of the N classes, Equation 1 can be extended as follows:

$$R_{H_N} = C \circ P \circ (L \cup (\bigcup_{i=1}^{N} L_{u_i} \circ G_{u_i} \circ T_{u_i}))^* \circ G_N \qquad (2)$$

where in this formulation, R_{H_N} represents the collection of $N+1$ search networks: the word-level IV search network and N subword search networks, each corresponding to a class of OOV words. The i^{th} subword search network is represented by $L_{u_i} \circ G_{u_i} \circ T_{u_i}$. The word level n-gram G_N includes the N different classes of OOV words. This n-gram can either use a class-specific language model probability or can use the same estimate for all classes. In our experiments, we explore various combinations of class-specific networks and word-level n-grams. Next, we describe two techniques for designing OOV classes.

3.1. Part-Of-Speech OOV Classes

Class assignments in terms of POS classifications can be used to design the multi-class OOV model. Starting with a tagged dictionary, words can be broken down into multiple classes. For training the word-level language model G_N, each OOV word in the training corpus is replaced with its POS tag, hence class-specific n-grams can be estimated. The subword-level language models, G_{u_i} can be trained on the phone sequences of all words belonging to this class of words. In designing OOV classes, we only use a small number of POS tags since many of the POS tags correspond to words that are not typical OOV words, such as function words. In order to resolve the problem of words belonging to multiple classes such as words that can be either verbs or nouns, we create intersection classes for POS tags that have significant overlap. For example words that can be either nouns or verbs, such as the word *book*, will belong to the class *noun-verb*.

3.2. Automatically-Derived OOV Classes

The second approach relies on a two-step clustering procedure. Given a list of words, the goal is to break the list down into N lists, one for each of the N classes. The first step is the initialization step. The goal of this first step is to obtain a good initial class assignment for each of the words. The second step is an iterative clustering procedure intended to move words from one class to another in order to minimize the overall model perplexity.

3.2.1. Step 1: Agglomerative Clustering

Agglomerative clustering is a bottom-up hierarchical clustering technique that starts by assigning each data point its own cluster. Based on some similarity measure, clusters are successively merged to form larger clusters. The process is repeated until the desired number of clusters is obtained [4]. The procedure uses a similarity measure that is based on the phonetic similarity of words. Given the phone sequences of two words w_i and w_j, the similarity measure $d(w_i, w_j)$ is the *phone-pair* edit distance between the two words. This distance is the minimum number of phone pair substitutions, deletions and insertions needed to match one word to another. This similarity measure groups words with similar phone pairs within the same cluster. Given the distance measure between individual words, we use an average similarity measure at the cluster level. At each step of the clustering procedure, we select for merging the pair of clusters X_m and X_n such that the average distance $d_{avg}(X_m, X_n)$ is minimum:

$$d_{avg}(X_m, X_n) = \frac{1}{c_m c_n} \sum_{w_i \in X_m} \sum_{w_j \in X_n} d(w_i, w_j) \qquad (3)$$

where c_m and c_n are the number of words in clusters X_m and X_n respectively. Because of the high computational requirements of this type of clustering, we run this step only on a randomly-chosen subset of the large dictionary of words.

3.2.2. Step 2: Perplexity Clustering

Given the classes from Step 1, we create a class-specific phone bigram language model for each class. Step 2 uses an iterative optimization technique similar to K-means clustering [4]. The basic idea is to move words from one class to another if such a move improves the value of some criterion function. For the OOV model, the criterion function we use is the word's phone sequence perplexity against the various n-grams. The procedure is repeated until the change in average perplexity is smaller than some threshold or no more words change classes.

A variation on the two-step automatic approach is to use the POS tags for initialization. Instead of performing the agglomerative clustering to initialize the word classes, we can start with the assignments from the POS tags and then perform the perplexity clustering described above. There are two advantages for such an approach. First, the initial assignment, being based on POS tags, could provide for a better starting point for the perplexity clustering. The second advantage is eliminating the computational overhead of agglomerative clustering required in Step 1.

3.3. Related Work

The only work we are aware of on the use of multi-class models for OOV recognition is the approach presented in [5]. In this work, Gallwitz *et al.* constructed five word categories that included cities, regions, and surnames. In addition, they defined a category for rare words that are not in the first five, as well as one for garbage words such as word fragments. Unlike our approach, they used very simple acoustic models for each of the OOV categories: a flat model that consisted of a fixed number of HMM states with identical probability density functions.

4. EXPERIMENTS AND RESULTS

All of the experiments for this work are within the JUPITER weather information domain [3]. A set of context-dependent diphone acoustic models was used, whose feature representation was based on the first 14 MFCCs averaged over 8 regions near hypothesized phonetic boundaries. Diphones were modeled using mixtures of diagonal Gaussians with a maximum of 50 Gaussians per model. The word lexicon consisted of a total of 2,009 words, many of which have multiple pronunciations. The training set consisting

of 88,755 utterances was used to train both the acoustic and the language models. The test set consisted of 2,029 utterances, 314 of which contained OOV words (most of the OOV utterances had only one OOV word). For the baseline OOV model, we used the dictionary configuration [2] where the subword lexicon is simply the phoneme set and the subword bigram is trained on phoneme sequences of all words in the LDC PRONLEX dictionary. PRONLEX contains 90,694 words with 99,202 unique pronunciations.

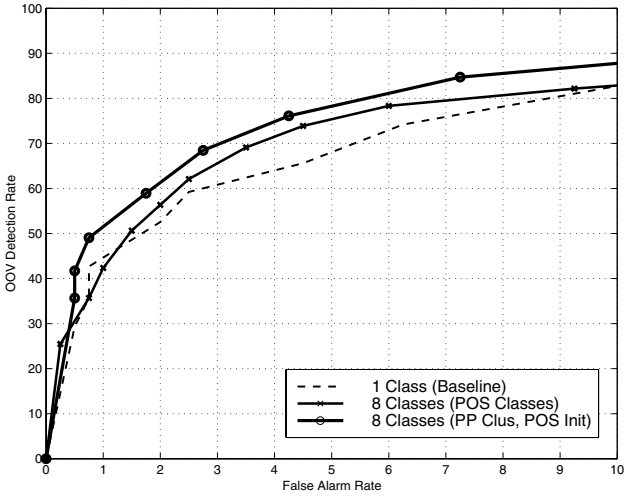

Fig. 1. ROC curves for the three OOV models discussed.

The behavior of the OOV model was measured by observing the OOV detection and false alarm rates on the test set as the cost of entering the OOV model C_{OOV} was varied, thereby obtaining the receiver operating characteristic (ROC) over a range of OOV detection and false alarm rates. Figure 1 shows the ROC curves for the three different models: a baseline single-class model, the POS eight-class model, and the automatically derived model. We will discuss each of the curves in the following sections. In order to quantify the ROC behavior, a *figure of merit* (FOM) was computed which measured the area under the ROC curve over the 0% to 10% false alarm rates. For our work we are most interested in the ROC region with low false alarm rates, since this produces a small degradation in recognition performance on IV data.

4.1. The POS Model

In order to get the POS tags of words in PRONLEX, we used the related dictionary COMLEX which contains a total of 22 POS tags. The majority of the words in PRONLEX belong to one of five main classes: nouns, verbs, adjectives, adverbs, and names. However, a significant overlap exists between the noun and verb classes, as well as between the adjective and verb classes. For our POS OOV model, we chose a model with eight classes: the five classes above, the two intersection classes noun-verb and adjective-verb and a backup class that covers OOV words that either are untagged or do not belong to any of the other seven classes. To build the eight OOV models, we use the phone-level lexicon for all eight classes. For each class we train its phone bigram using phone sequences of words that belong to the class.

Table 1 shows the detection results for the POS multi-class model. The multi-class extension can be done in one of three ways: (1) only at the language model level by having multiple OOV n-grams, (2) only at the OOV model level by having multiple OOV networks, one for each class, (3) both at the language model and the OOV model level. The table shows the three possible cases as well as the baseline. The first result in the table is the baseline system with an FOM of 0.64. The second case involve using the eight OOV classes for language modelling, but still using the same OOV model for all classes, i.e. using G_8 and one OOV network. The FOM for this condition is 0.65, only slightly better than the baseline. The third case involves creating multiple OOV networks but using the same language model n-grams for all classes. The FOM for this case is 0.68, a significant improvement over the baseline single class model. Adding the language model classes to this configuration does not improve performance. This is the fourth case in the table, where the FOM stays at 0.68.

Condition	G_1 **n-gram**	G_8 **n-gram**
1 OOV network	0.64	0.65
8 OOV networks	0.68	0.68

Table 1. FOM detection results on different configurations of the POS model.

The FOM results show that the improvement from the POS multi-class model is due mainly to using multiple OOV networks and not multiple word n-gram OOV classes. This finding could be specific to the JUPITER domain since it is a fairly simple recognition task where most of the OOV words are either names or weather terms (nouns). Hence the benefit from the OOV neighboring context is limited and does not help improve performance. Large vocabulary unconstrained domains may benefit more from multiple OOV n-gram classes. An important aspect of the POS model is its ability to identify the type or POS tag of the OOV word. A manual examination of the correctly detected OOV words showed that 81% of the detected OOV words are recognized with the correct POS tag.

The impact of the multi-class model on the word error rate (WER) is similar to that of a single class model. For all reported configurations, the relationship between overall OOV false alarm rate and WER on IV test data is approximately linear. The WER increases slowly from the baseline WER of 10.9% at 0% false alarm rate to under 11.5% at 10% false alarm rate.

4.2. The Automatically-Derived Model

For Step 1 in this approach, agglomerative clustering was done on 1,000 randomly chosen words from PRONLEX to produce 8 initial classes. Figure 2 shows the change in the weighted average perplexity of the multi-class model as a function of the iteration number for two initial conditions: the clusters obtained from agglomerative clustering in Step 1, and the clusters based on the POS tags. The clustering was iterated until the change in perplexity was less than 0.05. At that point, very few words moved from one class to another. Figure 2 shows that the multi-class model perplexity improved from 12.5 to 10.2 for the POS initialization and from 13.2 to 10.3 for the agglomerative clustering initialization.

In these experiments, we use a word n-gram with a single OOV class. There are two reasons for using multiple classes. The first is the fact that we did not get much gain from multiple n-

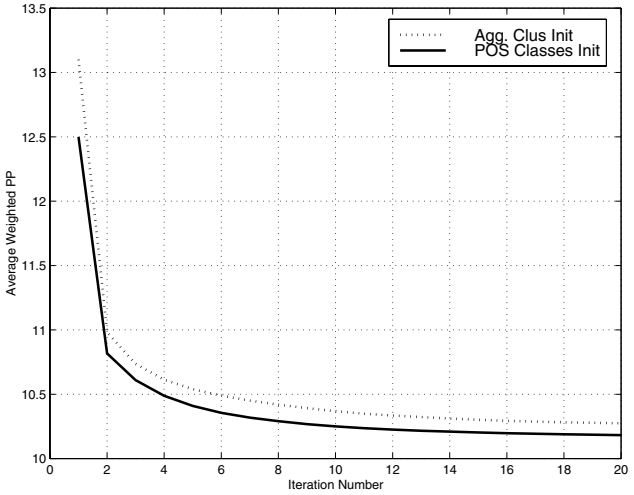

Fig. 2. Weighted average perplexity of the multi-class model in terms of the clustering iteration number.

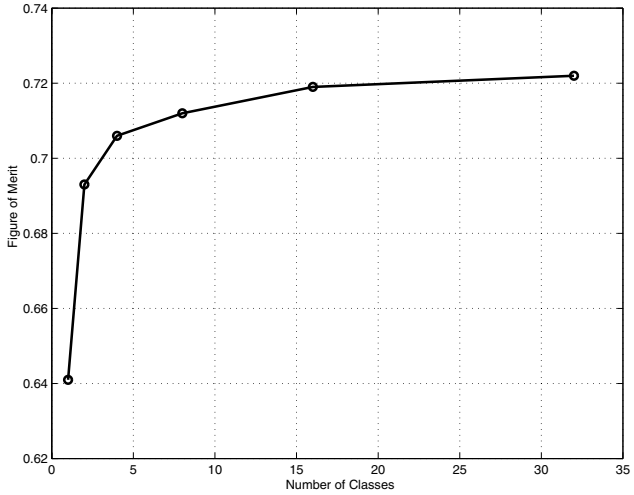

Fig. 3. The FOM performance of the automatic model as a function of the number of classes.

OOV Model	Number of Classes	FOM
Baseline	1	0.64
POS	**8**	**0.68**
PP Clus (AggClus Init)	**8**	**0.71**
PP Clus (POS Init)	**8**	**0.72**

Table 2. FOM detection results for various multi-class models.

gram OOV classes with the POS model. The second reason is that while the words in these automatically derived classes may be phonetically similar because of the way they are derived, they do not necessarily share similar contexts.

Detection results are summarized in Table 2 and Figure 1. As shown, the automatic OOV model with the POS initialization outperforms the single class model as well the POS model. Note that using POS tags for initialization is only slightly better than using agglomerative clustering. Over the baseline system using one class, the multi-class model improves the FOM by over 11% (from 0.64 to 0.72). For example, at an OOV detection rate of 70%, the false alarm rate is reduced from 5.3% for a single class to 2.9% for an eight-class model. The FOM of 0.72 for the multi-class model with POS initialization is also better than the 0.70 FOM result we reported in [2] with the mutual information approach where a single OOV model was used but with multi-phone sublexical units.

4.2.1. Varying the Number of Classes

Figure 3 shows the performance of the automatic multi-class model as a function of the number of classes N. We compared using 1, 2, 4, 8, 16, and 32 classes. Figure 3 shows that most of the gain is obtained in going from one to two classes where the FOM jumps from 0.64 to over 0.69. The benefit from using more classes diminishes as N increases. Going from 8 to 16 and then to 32 classes gives only a slight improvement in the FOM. This behavior could be specific to JUPITER, and other unconstrained domains may benefit more from a larger number of classes.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a multi-class extension to our approach for modelling OOV words. Instead of augmenting the word search network with a single OOV model, we added several OOV models, each corresponding to a class of OOV words. We presented two approaches for designing the OOV classes. The first approach relies on using common POS tags to design the OOV classes, while the second approach uses a two-step clustering procedure. The experimental results showed significant improvement over using the single class model reported earlier [1, 2].

In future work we plan to explore combining the multi-class approach with using multi-phone units within each OOV network. We also plan to explore using our approach for detecting out-of-domain utterances. Finally, we plan to investigate using a second-stage search with a large off-line dictionary to determine the identity of the OOV words after they are detected.

6. REFERENCES

[1] I. Bazzi and J. Glass, "Modelling out-of-vocabulary words for robust speech recognition," in *Proc. Intl. Conf. on Spoken Language Processing*, Beijing, Oct. 2000, pp. 401–404.

[2] I. Bazzi and J. Glass, "Learning units for domain-independent out-of-vocabulary word modelling," in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Sept. 2001, pp. 61–64.

[3] V. Zue, et al., "JUPITER: A telephone-based conversational interface for weather information," *IEEE Trans. on Speech and Audio Processing*, 88(1), 2000.

[4] R. Duda, and P. Hart, "Pattern Classification and Scene Analysis," *John Wiley & Sons* New York, 1973.

[5] F. Gallwitz, E. Noeth, and H. Niemann (1996). "A category based approach for recognition of out-of-vocabulary words." in *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, Oct. 1996, pp. 228–231.