FROM WORD-SPOTTING TO OOV MODELING

Paul Fitzpatrick (6345g11)

6.345, Automatic Speech Recognition, Spring 2001

1 Introduction

This paper explores one dimension along which word spotting and speech recognition differ: the nature of the background model. In word spotting, a relatively small number of keywords float on a sea of unknown words. In speech recognition, an occasional unknown word punctuates utterances that are otherwise completely in-vocabulary. Despite this difference in viewpoint, in some circumstances implementations of the two may become very similar. When transcribed data is available for a domain, word spotting benefits from the more detailed background model this can support [9]. The manner in which the background is modeled in these cases is reminiscent of speech recognition. For example, a large vocabulary with good coverage may be extracted from the corpus, so that relatively few words in an utterance remain unmodeled. In this case, the situation is qualitatively similar to OOV modeling in a conventional speech recognizer, except that the vocabulary is strictly divided into "filler" and "keyword".

This paper describes a mechanism for bootstrapping from a relatively weak background model for word-spotting, where OOV words dominate, to a much stronger model where many more word or phrase clusters have been "moved to the foreground" and explicitly modeled. With this increase in vocabulary comes an increase in the potency of language modeling, boosting performance on the original vocabulary.

The following sections show how a conventional speech recognizer can be convinced to cluster frequently occurring acoustic patterns, without requiring the existence of transcribed data.

2 Boot-strapping the lexicon

A recognizer with a phone-based OOV model is able to recover an approximate phonetic representation for words or word sequences that are not in its vocabulary. If commonly occurring phone sequences can be located, then adding them to the vocabulary will allow the language model to capture their co-occurrence with words in the original vocabulary, potentially boosting recognition performance. This suggests building a "clustering engine" that scans the output of the speech recognizer, correlates OOV phonetic sequences across all the utterances, and updates the vocabulary with any frequent, robust phone sequences it finds. While this is feasible, the kind of judgments the clustering engine needs to make about acoustic similarity and alignment are exactly those at which the speech recognizer is most adept. This section describes a way to convince the speech recognizer to perform clustering almost for free, eliminating the need for an external module to make acoustic judgments.

The clustering procedure is shown in Figure 1. An *n*gram-based language model is initialized randomly, or trained up using whatever data is available – for example, a small collection of transcribed utterances. Unrecognized words are explicitly represented using a

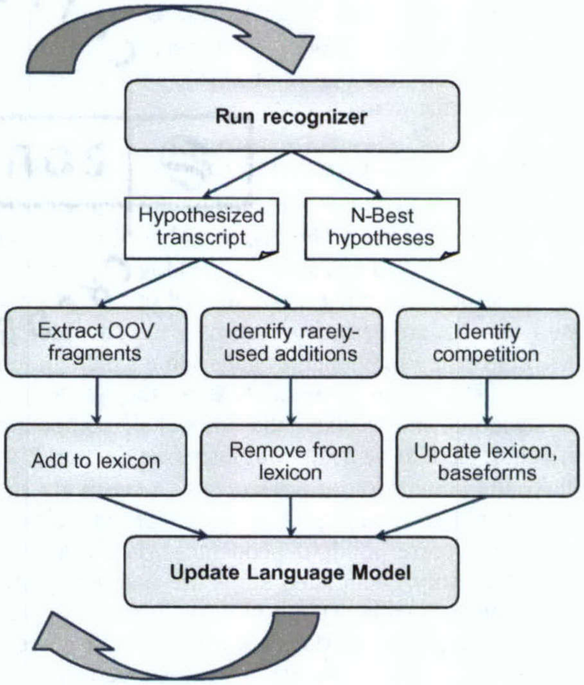

Figure 1: The iterative clustering procedure.

phone-based OOV model, described in the next section. The recognizer is then run on a large set of untranscribed data. The phonetic and word level outputs of the recognizer are compared so that occurrences of OOV words are assigned a phonetic transcription. A randomly cropped subset of these are tentatively entered into the vocabulary, without any attempt yet to evaluate their significance (e.g. whether they occur frequently, whether they are dangerously similar to a keyword, etc.). The hypotheses made by the recognizer are used to retrain the language model, making sure to give the newly added vocabulary items some probability in the model. Then the recognizer runs using the new language model and the process iterates. The recognizer's output can be used to evaluate the worth of the new "vocabulary" entries. The following sections detail how to eliminate vocabulary items the recognizer finds little use for, and how to detect and resolve competition between similar items.

2.1 Extracting OOV phone sequences

The recognizer used the OOV model described in [1], contributed by Issam. This model can match an arbitrary sequence of phones, and has a phone bigram to capture phonotactic constraints. The OOV model is placed in parallel with the models for the words in the vocabulary. A cost parameter can control how much the OOV model is used at the expense of the in-vocabulary models. This value was fixed at zero throughout the experiments described in this paper, since it was more convenient to control usage at the level of the language model. The bigram used in this project is exactly the one used in [1], with no training for the particular domain.

2.2 Recovering phonemic representations

It is useful to convert the extracted phone sequences to phonemes if they are to be added as baseforms in the lexicon. Although the sequences could be kept in their original form by creating a dummy set of units for the baseforms that are passed verbatim by the phonological rules, converting to phonemes adds some small amount of generalization over allophones to the sequence's pronunciation, and reduces the amount of competing forms that have to be dealt with later (see Section 2.4). I make the conversion in a naïve way, classifying single or paired phonetic units into a set of equivalence classes that correspond to phonemes. For example, taps and cleanly enunciated stops are mapped to the same phoneme, with explicit closures being dropped. Although the procedure does not capture some contextual effects, it achieves perfectly adequate performance (see Section 3).

Phoneme sequences are given an arbitrary name and added to the list of vocabulary and baseforms. To ensure that the language model assigns some probability to these new vocabulary items the next time the recognizer runs, a collection of randomly generated sentences is added to those output of the recognizer used in re-training.

2.3 Dealing with rarely-used additions

If a phoneme sequence introduced into the vocabulary is actually a common sound sequence in the acoustic data, then the recognizer will pick it up and use it. Otherwise, it just will not appear very often in hypotheses. After each iteration a histogram of phoneme sequence occurrences in the output of the recognizer is generated, and those below a threshold are cut.

2.4 Dealing with competing additions

Very often, two or more very similar phoneme sequences will be added to the vocabulary. If the sounds they represent are in fact commonly occurring, both are likely to prosper and be used more or less interchangeably by the recognizer. This is unfortunate for language modeling purposes, since their statistics will not be pooled and so will be less robust. Happily, the output of the recognizer makes such situations very easy to detect. In particular, this kind of confusion can be uncovered through analysis of the N-best utterance hypotheses.

If we imaging a set of N-best hypotheses aligned and stacked vertically, then competition is indicated if two vocabulary items exhibit both of these properties:

- Horizontally repulsive – if one of the items appears in a single hypothesis, the other will not appear in its vicinity.
- Vertically attractive – the items frequently occur in the same part of a collection of hypotheses for a particular utterance.

Since the utterances in this domain are generally short and simple, it did not prove necessary to rigorously align the hypotheses. Instead, items were considered to be aligned based simply on the vocabulary items preceding and succeeding them. It is important to measure both the attractive and repulsive conditions to distinguish competition from vocabulary items that are simply likely or unlikely to occur in close proximity.

Accumulating statistics about the above two properties across all utterances gives a reliable measure of whether two vocabulary items are essentially acoustically equivalent to the recognizer. If they are, they can be merged or pruned so that the statistics maintained by the language model will be well trained. For clear-cut cases, the competing items are merged as alternatives in the baseform entry for a single vocabulary unit. A better alternative might have been to use class n-grams and put the items into the same class, but this works fine. For less clear-cut cases, one item is simply deleted.

Here is an example of this process in operation in the very first iteration of the algorithm after new vocabulary items have been added. These are the 10-best hypotheses for the given utterance:

"what is the phone number for victor zue"

```
<oov> phone (n ah m b er) (m ih t er z) (y uw)
<oov> phone (n ah m b er) (m ih t er z) (z y uw)
<oov> phone (n ah m b er) (m ih t er z) (uw)
<oov> phone (n ah m b er) (m ih t er z) (z uw)
<oov> phone (ah m b er f) (m ih t er z) (z y uw)
<oov> phone (ah m b er f) (m ih t er z) (y uw)
<oov> (ax f aa n ah) (m b er f axr) (m ih t er z) (z y uw)
<oov> (ax f aa n ah) (m b er f axr) (m ih t er z) (y uw)
<oov> phone (ah m b er f) (m ih t er z) (z uw)
<oov> phone (ah m b er f) (m ih t er z) (uw)
```

The "<oov>" symbol corresponds to an out-of-vocabulary sequence. The phone sequences within parentheses are uses of items added to the vocabulary in the last iteration. From this single utterance, we acquire evidence that:

- The entry for (ax f aa n ah) may be competing with the keyword "phone". If this holds up statistically across all the utterances, the entry will be destroyed. The keyword vocabulary is given special status, since they represent a link to the outside world that should not be modified.
- (n ah m b er), (m b er f axr) and (ah m b er f) may be competing. They are compared against each other because all of them are followed by the same sequence (m ih t er z) and many of them are preceded by the same word "phone".
- (y uw), (z y uw), and (uw) may be competing

All of these will be patched up for the next iteration. Section 3 shows stable baseforms created through this process.

This use of the N-best utterance hypotheses is reminiscent of their application to computing a measure of recognition confidence in [3].

2.5 Testing for convergence

For any iterative procedure, it is important b know when to stop. If we have transcribed data, we can track the keyword error rate on that data and halt when the increment in performance is sufficiently small.

If there is no transcribed data, then we cannot directly measure the error rate. We can however bound the rate at which it is changing by comparing keyword locations in the output of the recognizer between iterations. If few keywords are shifting location, then the error rate cannot be changing above a certain bound. We can therefore place a convergence criterion on this bound rather than on the actual keyword error rate. It is important to just measure changes in keyword locations, and not changes in vocabulary items added by clustering. Items that do not occur often tend to be destroyed and rediscovered continuously, making comparisons difficult.

3 Qualitative Results

This section describes, through examples, the kinds of vocabulary discovered by the clustering procedure. Numerical, performance-related results are reported in Section 4.

Results given here are from a clustering session with an initial vocabulary of five keywords (`email`, `phone`, `room`, `office`, `address`), run on the training data, and not using the transcripts for that data at all.

Here are the top 10 clusters discovered on this very typical run, ranked by decreasing frequency of occurrence:

1	n ah m b er	6	p l iy z
2	w eh r ih z	7	ae ng k y uw
3	w ah t ih z	8	n ow
4	t eh l m iy	9	hh aw ax b aw
5	k ix n y uw	10	g r uw p

These clusters are used consistently by the recognizer in places corresponding to: "number, where_is, what_is, tell_me, can_you, please, thank_you, no, how_about, group," respectively. The first, /n ah m b er/, is very frequent because of "phone number", "room number", and "office number". Once it appears as a cluster the language model is immediately able to improve recognition performance on those keywords.

The word groups picked out are actually rather like the merged words often placed in a conventional lexicon – "where_is", "what_is" etc.

Other high-frequency clusters correspond to common first names (Karen, Michael). Victor Zue, /ih t er z uw/, and Jim Glass, /jh ih n b ae s/, get clusters all to themselves. Note the loss of the initial fricative in Victor – this is typical (see also the rendering of thank_you as /ae ng k y uw/). This may be partially due to the characteristics of speech over a phone line, where much of the high frequency component is lost. The remaining clusters are less likely to correspond to anything meaningful and have little effect on recognition performance. Parts of people's names are common.

Curiously the cluster corresponding to yes, /y eh s/, consistently takes longer to appear and is lower in frequency than no, /n ow/, which is very frequent. Possibly people were saying "no!" to the early phone-in recognizer much more than they were saying "yes!"

Every now and then a "parasite" appears such as /dh ax f ow n/ (from an instance of "the phone" that the recognizer fails to spot) or /iy n eh l/ (from "email"). These have the potential to interfere with the detection of the keywords they resemble acoustically. But as soon as they have any success, they are detected and eliminated as described in Section 2.4. It is possible that if a parasite doesn't get greedy, and for example limits itself to one person's pronunciation of a keyword, that it will not be detected, although I didn't see any examples of this happening.

Many simple sentences can be modeled completely after clustering, without need to fall back on the generic OOV phone model. For example, the utterances:

What is Victor Zue's room number
Please connect me to Leigh Deacon

are recognized as:

(w ah t ih z) (ih t er z uw) room (n ah m b er)
(p l iy z) (k ix n eh k) (m iy t uw) (l iy d iy) (k ix n)

All of which are entries in the vocabulary and so contribute to the language model. All the discovered vocabulary items are assigned one or more baseforms as described in Section 2.4. These baseforms often cover trivial variations in a feature of one or two phones. For example, following the format of the baseforms file we have:

```
t_eh_l_m_iy:   ( t eh l m iy ,    d eh l m iy )
p_l_iy_z:      ( p l iy z ,       p l iy s )
w_er_k:        ( w er k ,         w ao r k )
```

Other baseforms contain more variation:

```
n_ah_m_b_er:   ( n ah m b er ,    ah m b er ,
                 en ah m b er )
w_ah_t_ih_z:   ( w ah t ih z ,    w a h d ih z ,
                 w ah t s ,       w ah t s t ,
                 w ah t er ,      w ah s dh ax )
```

The nasal in /n ah m b er/ is sometimes recognized, sometimes not, so both pronunciations are added to a single baseform. Short, often unstressed words such as the definite and indefinite articles are not clustered by the algorithm. Their influence instead appears in baseforms, for example the /w ah s dh ax/ entry above.

4 Quantitative Results

For experiments involving small vocabularies, it is appropriate to measure performance in terms of Keyword Error Rate (KER). I take this to be:

$$KER = \left(\frac{F+M}{T} \times 100 \right)\% \text{ , with:}$$

F : Number of false or poorly localized detections
M : Number of missed detections
T : True number of keyword occurrences in data

A detection is only counted as such if it occurs at the right time. Specifically, the midpoint of the hypothesized time interval must lie within the true time interval the keyword occupies. I take forced alignments of the test set as ground truth. This means that for testing it is better to omit utterances with artifacts and words outside the full vocabulary, so that the forced alignment is likely to be sufficiently precise.

The experiments here are designed to identify when clustering leads to reduced error rates on a keyword vocabulary. Since the form of clustering addressed in this paper is fundamentally about extending the vocabulary, we would expect it to be useless if the vocabulary is already large enough to give good coverage. We would expect it to offer the greatest improvement when the vocabulary is smallest. To measure the effect of coverage, the full vocabulary was made smaller and smaller by incrementally removing the most infrequent words. A set of keywords were chosen and kept constant and in the vocabulary across all the experiments so the results would not be confounded by properties of the keywords themselves (for example, the most common word "the" would make a very bad keyword since it is often unstressed and loosely pronounced). The same set of keywords were used as in Section 3.

Clustering is again performed without making any use of transcripts. To truly eliminate any dependence on the transcripts, an acoustic model trained only on Pegasus data was used. This reduced performance but made it easier to interpret the results.

Figure 2 show a plot of error rates on the test data as the size of the vocabulary is varied to provide different degrees of coverage. The most striking result is that the clustering mechanism reduces the sensitivity of performance to drops in coverage. In this scenario, the error rate achieved with the full vocabulary (which gives 84.5% coverage on the training data) is 33.3%. When the coverage is low, the clustered solution error rate remains under 50% — in relative terms, the error increases by at most a half of its best value. Straight application of a language model gives error rates that more than double or treble the error rate.

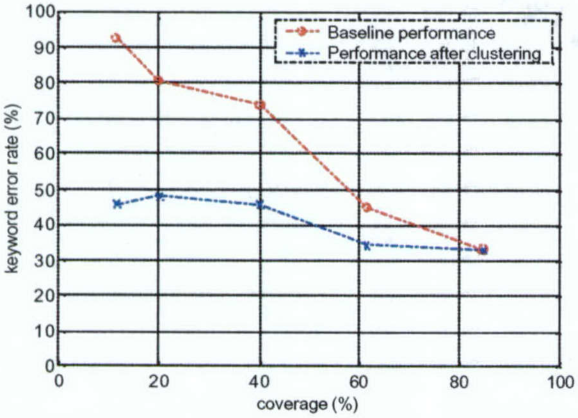

Figure 2: Keyword error rate of baseline recognizer and clustering recognizer as total coverage varies.

As a reference point, the keyword error rate using a language model trained with the full vocabulary on the full set of transcriptions with an acoustic model trained on all available data gives an 8.3% KER.

5 Conclusions

Speech recognizers are painstakingly engineered to factor all available data into making judgments of acoustic similarity. This makes them a natural tool for clustering acoustic data that is hard to beat (and I tried). A recognizer that generates N-best hypotheses is particularly suited to this task.

The clustering mechanism described in this paper can build a language model based on untranscribed data. In the interval between the start of acoustic data collection and the point at which enough data has been transcribed to provide reasonable coverage, clustering has the potential to boost performance. This might be useful in off-the-shelf systems designed for non-experts, so that the user sees a quicker return on their efforts.

An important issue not touched on at all here is whether it is possible to train an acoustic model from untranscribed data. This seems a much harder problem. But in the low-coverage regime clustering is aimed at, the language model is likely to be the limiting factor to performance.

Acknowledgements

Many thanks to Issam Bazzi for his OOV model.

References

[1] Bazzi, J.R. Glass, *Modeling Out-of-Vocabulary Words for Robust Speech Recognition*, Proc. 6th International Conference on Spoken Language Processing, Beijing, China October 2000.

[2] Bazzi, J.R. Glass, *Learning Units for Domain-Independent Out-of-Vocabulary Word Modelling.* Not yet published.

[3] T.J. Hazen, I. Bazzi, *A Comparison and Combination of Methods for OOV Word Detection and Word Confidence Scoring*, Proceedings of the 2001 International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, May, 2001.

[4] Manos, *A Study on Out-of-Vocabulary Word Modeling for a Segment-based Keyword Spotting System*, Masters Thesis, MIT 1996.

[5] R. Gretter, G. Riccardi, *On-line Learning of Language Models with Word Error Probability Distributions*, Proc. International Conference on Acoustics, Speech, and Signal Processing, Utah, 2001.

[6] G. Riccardi, S. Bangalore, P.D. Sarin, *Learning Head-Dependency Relations from Unannotated Corpora*, Proc. IEEE Automatic Speech Recognition and Understanding Workshop, Colorado, 1999.

[7] A.L. Gorin, D. Petrovksa-Delacrétaz, G. Riccardi, J.H. Wright, *Learning Spoken Language without Transcriptions*, Proc. IEEE Automatic Speech Recognition and Understanding Workshop, Colorado, 1999.

[8] M.R. Brent, J.M. Siskind, *The Role of Exposure to Isolated Words in Early Vocabulary Development*, Cognition, to appear.

[9] P. Jeanrenaud, K. Ng, M. Siu, J.R. Rohlicek, H. Gish, *Phonetic-based Word Spotter: Various Configurations and Application to Event Spotting*, Proc. EUROSPEECH '93.

[10] E.D. Sandness, I.L. Hetherington, *Keyword-based Discriminative Training of Acoustic Models*, Proc. 6th International Conference on Spoken Language Processing, Beijing, China 2000.

[11] G. Riccardi, A.L Gorin, *Stochastic Language Adaptation over Time and State in Natural Spoken Dialogue Systems*, IEEE Transactions on Speech and Audio Processing, Vol 8(1), January 2000.

[12] G. Chung, S. Senaff, L. Hetherington, *Towards Multi-Domain Speech Understanding Using a Two-Stage Recognizer*, Proc. Eurospeech 99, Budapest, Hungary, September 1999.

[13] Bazzi, J.R. Glass, *Heterogeneous Lexical Units for Automatic Speech Recognition: Preliminary Investigations*, Proc. International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, June 2000.