A Two-pass Strategy for Handling OOVs in a Large Vocabulary Recognition Task

Odette Scharenborg[1] *and Stephanie Seneff*[2]

[1]CLST, Radboud University Nijmegen, Nijmegen, The Netherlands
[2]Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge, MA, USA
o.scharenborg@let.ru.nl, seneff@csail.mit.edu

Abstract

This paper addresses the issue of large-vocabulary recognition in a specific word class. We propose a two-pass strategy in which only major cities are explicitly represented in the first stage lexicon. An *unknown word* model encoded as a phone loop is used to detect OOV city names (referred to as rare city names). After which SpeM, a tool that can extract words and word-initial cohorts from phone graphs on the basis of a large fallback lexicon, provides an *N*-best list of promising city names on the basis of the phone sequences generated in the first stage. This *N*-best list is then inserted into the second stage lexicon for a subsequent recognition pass.

Experiments were conducted on a set of spontaneous telephone-quality utterances each containing one rare city name. We tested the size of the *N*-best list and three types of language models (LMs). The experiments showed that SpeM was able to include nearly 85% of the correct city names into an *N*-best list of 3000 city names when a unigram LM, which also boosted the unigram scores of a city name in a given state, was used.

1. Introduction

This paper addresses the issue of large-vocabulary recognition for a specific class of words, in the context of telephone-access spoken dialogue systems. The practical interest of this work is illustrated using two on-line systems which offer flight (Mercury, [1]) and weather (Jupiter, [2]) information worldwide. The weather source has recently been updated to handle over 38,000 city names (hereafter referred to as 'rare' city names). The flights system would also be able to handle any city that has an airport, if it only could recognize the city name. A big issue, then, is how to handle a large set of city names effectively and efficiently in the speech recognizer. A straightforward strategy is to expand the recognizer's lexicon. However, this will result in a large search space, with only a back-off prior probability associated with each of the rare city names. Very large lexicons in combination with a weak language model (LM) usually results in poor performance for automatic speech recognition (ASR) systems.

In this paper, we propose a two-stage ASR system. To overcome the problem of a weak language model, we adopt a novel strategy that uses small lexicons in combination with a generic phone-based *out-of-vocabulary (OOV) word* model to represent a rare city name in the form of a phone graph. This approach licenses in a second stage only those city names that match the proposed phone graph sufficiently well.

In the literature, a variety of solutions to handle OOV words have been proposed, e.g. [3],[4]. In [5], in accordance with [3], we built a two-stage recognizer that detects OOV intervals in the first stage, and that adapts the lexicon of the second stage recognizer by selecting a subset from a large *fallback* lexicon, which in our case consists of city names. To select the subset of city names from the fallback lexicon, we use SpeM [6]. The aim of the second stage is to recognize as many of the rare city names that were marked as OOV by the first stage recognizer as possible. Since an ASR system can only recognize those words that are included in its lexicon, it is clear that the performance of the second stage recognizer on recognizing the OOV words is crucially dependent on whether the correct word is included in the second stage recognizer's lexicon. Optimizing the coverage of the second stage lexicon is the main focus of this work.

Initial experiments with the proposed two-pass system were presented in [5]. SpeM was able to select nearly 60% of the correct rare city names (in 399 utterances) from a fallback lexicon containing 52,595 city names in an *N*-best hypothesis list of 3000 city names. In those experiments, no language model for SpeM was available: All words in the fallback lexicon had equal probability. It was suggested that it might be possible to improve the performance of SpeM and the two-pass recognition system by using population statistics (in the form of unigram counts) as unigram scores for the city names. In this paper, we put this suggestion to the test.

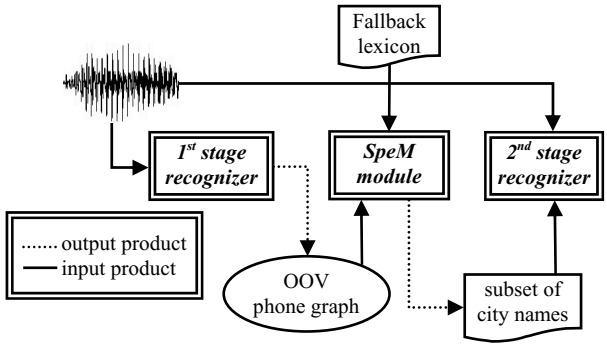

Figure 1. Overview of proposed the two-stage recognition system.

2. Recognition system

The proposed two-stage recognition system is schematically depicted in Figure 1. The acoustic signal is fed into the first stage recognizer, which uses a lexicon that captures 'general' words in addition to the 500 most frequent city names. An OOV model that is intended to mark all city names not in the lexicon as being OOV is integrated into the first stage. The hypothesized phone graphs underlying the stretches of speech signal marked as OOVs can be extracted. These OOV phone graphs are used by the SpeM module to select the most likely city names from the fallback lexicon for that specific

utterance. This subset of most likely city names is then added to the 'utterance-dependent' lexicon of the second stage. The second stage recognizer then does a new recognition on the basis of the same acoustic models as were used in the first stage.

2.1. The automatic speech recognizer

The two-stage recognizer used in this study is the segment-based speech recognition system SUMMIT [7], which uses Finite State Transducers (FSTs) to represent its search space.

The procedure used to mark the OOV words and generate the OOV phone graphs is described in detail in [4]: The *generic word model* is implemented as a phone loop that allows for phone sequences of arbitrary length. This OOV model is included in the lexicon. The transition into the generic word model is controlled via an OOV penalty. This OOV penalty can be considered as a unigram score: It controls how easily the OOV 'word' is selected.

For each utterance in which an OOV was hypothesized in the word lattice, only one OOV phone graph was generated (due to the current implementation of the procedure to extract the OOV phone graphs – this procedure can be adapted). Note that it is possible that the phone graph does not match exactly with the stretch of speech that contains the rare city, because preceding and trailing garbage phones may be present, or the phone graph may be cut off too early. Also, phone recognition errors in the city name itself can cause problems. Furthermore, it is possible that the first stage recognizer recognizes the rare city name as an in-vocabulary word or maps an in-vocabulary word on the OOV.

The recognizers in the first and second stage are identical, with the exception of the lexicon: The second stage recognizer also has a 'dynamic' lexicon [8] that is supplied with the list of rare city names extracted by SpeM from the fallback lexicon.

In the LMs, frequent and rare city names are treated as different classes and separate LM scores are calculated for them.

2.2. SpeM

SpeM was originally implemented to serve as a tool for research in the field of human speech recognition (HSR). It is a new and extended implementation of the theory underlying the *Shortlist* model, a computational model of human word recognition [9]. The main advance of SpeM over pre-existing computational models of HSR is that SpeM uses the acoustic speech signal as input, while Shortlist and other computational models of HSR only take handcrafted symbolic representations as input.

SpeM consists of two modules: An *automatic phone recognizer* (APR) and a *word search module*. The word search module parses the probabilistic phone graph created by the APR in order to find the most likely (sequence of) words, and computes for each word its activation based on the accumulated acoustic evidence for that word [6]. In the experiments described in this paper, the phone graphs are created by the first stage recognizer. In the remainder of this paper, whenever the word 'SpeM' is used, this actually only refers to the word search module of SpeM.

In SpeM, the sequence of words with the smallest phonemic distance between the sequence of phones on the path through the OOV phone graph and the phonemic representations of the words in the fallback lexicon is determined using a time-synchronous and breadth-first DP algorithm. Each phone insertion, deletion, and substitution is penalized according to independent penalties which can be tuned separately [6]. Furthermore, a garbage phone model is included in the lexicon. This garbage phone model is mapped onto phones appearing at the start and end of the phone graph that belong to the preceding or following word. The output of SpeM consists of an *N*-best list of hypothesized parses. Each parse contains words, word-initial cohorts (words sharing phone prefixes), garbage, silence, and any combination of these, except that a word-initial cohort can only occur as the last element in the parse. Thus, in addition to recognizing full words, SpeM is able to recognize partial words.

3. Material

The experiments were conducted on a set of continuous speech utterances, recorded from telephone conversations with the Jupiter and the Mercury system. The independent test set consisted of 241 utterances taken from both domains, each utterance containing exactly one rare city name.

The fallback city name lexicon used by SpeM contains 16,916 city names, which were harvested from the World Wide Web [10]. Many of the cities were non-existent in our lexical base forms resource file, and pronunciations were therefore automatically generated for them using the letter-to-sound system described in [8],[11]. The automatically generated pronunciations have been manually corrected, but errors may still remain.

The data sets used in these experiments are subsets of the data sets used in [5]. To investigate the influence of population statistics on the performance of our two-pass system, we only used those (utterances containing) city names for which population statistics were available.

The lexicon of the first stage consisted of the 'general' words from both domains, a list of the 500 most frequent city names, all US state names, and a set of 1,326 partial and short city names with a phonemic representation of three phones or less, such as 'los' and 'new' – this to simplify SpeM's task, since short words are difficult to find in a phone lattice. This resulted in a lexicon of 2,802 words.

Following [5], the lexicon of the second stage consisted of all words of a specific utterance in the 50-best list of that utterance created by the recognizer in the first stage, the 100 words that were most often deleted by the recognizer in the first stage (see also [5]), and the subset of most likely city names selected from the fallback lexicon by SpeM.

4. Experimental set-up

4.1. Extracting the subset from the fallback lexicon

In the first experiment, we tested two variables to improve the selection of the rare city names from the fallback lexicon:

- The size of the utterance-dependent *N*-best lists generated by SpeM.
- The effect of adding different types of LMs.

The results of these experiments are presented in terms of coverage: The percentage of the test set utterances for which the target rare city name (which was presumably marked as OOV by the first stage recognizer) is present in the *N*-best list generated for that utterance by SpeM.

4.1.1. The language models

In our unigram LM for SpeM, all identical city names are

mapped onto the same item and their unigram counts are summed. Because the first stage lexicon handles the frequent cities explicitly, we excluded their unigram counts from the calculations, but did not exclude the city names from the lexicon if they exist elsewhere. For example, 'Boston' exists in three states, and 'Boston, MA' is a 'frequent' city. Therefore, we compute the unigram score for 'Boston' by summing only 'Boston, GA' and 'Boston, IN'.

The reasoning behind the second type of LM is that if a city name is more likely on the basis of the context of the utterance, it should receive a higher probability. An obvious cue is the state name. It is highly likely that a city, which is uttered in the same utterance as a state name, lies in that state. To that end, we built utterance-dependent LMs for SpeM for those utterances in which a state name was present. If a state name is present in the *N*-best list generated by the first stage recognizer, all city names in that state receive a higher unigram score (identical to the one used in [5]). Of course, only those utterances in which a state name is present might benefit from this approach. This LM type is referred to as 'unigram+state'.

The performance of SpeM while using the two types of LMs is compared to the results of our baseline set-up in which an LM with equal probability for all city names was used to guide the search of SpeM ('zerogram LM').

4.2. The second stage

In the second experiment, the *N*-best lists generated by SpeM were included in the dynamic lexicon of the second stage recognizer. We examined the effect of varying sizes of the *N*-best list on the recognition performance of the full recognition system. An *N*-best list of 0 is used to serve as a baseline.

Furthermore, we compared a system in which all words in the dynamic list have equal probability ('Zerogram LM') with a system in which the unigram scores (see previous section) are added to the words in the dynamic list ('Unigram LM'). The results of this experiment are presented in terms of word accuracy and, since we are mainly interested in the recognition of the rare city names, the number of correctly recognized rare city names.

Table 1. Coverage results for varying sizes of the *N*-best lists generated by SpeM for the 241 utterances of the test set.

N	**Zerogram LM**		**Unigram LM**		**Unigram+state**	
	#	%	#	%	#	%
500	166	68.9	168	69.7	175	72.6
1000	180	74.7	179	74.3	184	76.3
1500	187	77.6	185	76.8	189	78.4
2000	192	79.7	190	78.8	193	80.1
2500	193	80.1	196	81.3	199	82.6
3000	194	80.5	200	83.0	203	84.2

5. Results

5.1. Extracting the subset from the fallback lexicon

The results of the first experiment are shown in Table 1. The column 'Zerogram LM' presents the results of the baseline set-up (all words have equal probability) in terms of absolute number of utterances for which the correct rare city name was present in the *N*-best list ('#') and as a percentage of the total number of 241 utterances of the test set ('%'). Likewise, the results for the set-up using the unigram LM and the set-up using the LM that boosted the counts of city names in a given state are shown in the columns 'Unigram LM' and 'Unigram+state', respectively.

5.1.1. Baseline: Zerogram LM

The coverage results of our baseline set-up show that already over 68% of the rare city names that were missing from the lexicon of the first stage recognizer are present in the lexicon of the second stage. This is an encouraging result, bearing in mind that all 16,916 words in the fallback lexicon have equal probability, and that the generated OOV graphs are far from perfect. Comparing the coverage for the *N*-best sizes 500 and 3000 clearly shows that increasing the length of the *N*-best list 6-fold does not increase the coverage proportionally. But still 28 more correct rare city names were present when the *N*-best list size was 3000. Note that the coverage in [5] for the 3000-best list was only 59.9%. The task SpeM faces in this experiment is easier than the task in [5], due to the smaller fallback lexicon.

5.1.2. Unigram LM

Using an LM that incorporates the unigram probabilities for the city names, on average improves the coverage, but the improvements are only small and do not occur for all sizes of the *N*-best list. The biggest improvement is obtained for the 3000-best list, while the performance slightly deteriorates for sizes 1000 through 2000. Analysis of the unigram counts shows that the mean unigram count of the city names in the test set is 14,836, while the mean unigram count of the city names in the fallback lexicon is 8,221. The deterioration in performance for sizes of *N* between 1000 and 2000 is thus not likely due to a low mean word count. Those words that were included in the *N*-best list when using the zerogram LM and that were no longer included when using the unigram LM most probably have a high number of confusable words in the fallback lexicon which, on top of that, probably have a larger population than the 'correct' city.

Table 2. Analysis of the utterances with/without a state name and for each how often the correct city name is (not) included in the 500-best list generated by SpeM.

	#utts
State name present in utterance:	**202**
• *State name present in output 1st stage recognizer:*	*171*
- City name present in 500-best list	116
- City name not present in 500-best list	55
• *State name not present in output 1st stage recognizer:*	*31*
- City name present in 500-best list	18
- City name not present in 500-best list	13
State name not present in utterance:	**39**
- City name present in 500-best list	32
- City name not present in 500-best list	7

5.1.3. Unigram+state LM

Next, we built utterance-dependent LMs for SpeM for those utterances in which a state name was present. To determine the maximum gain that can be obtained with this approach, we tabulated how many of the 241 utterances contained a state name, and how often the correct city name was included in the 500-best list generated by SpeM. Table 2 shows the details of this analysis: In 202 of the 241 utterances, a state name was present. The 39 utterances in which no state name was present will not benefit from adding utterance-dependent LMs. Looking closer at this set of utterances, however,

revealed that for 32 utterances of these 39 utterances, the city name was already present in the 500-best list.

For 171 of the 202 utterances for which a state name was present in the utterances, the first pass recognizer found a state name. Of these 171 utterances, the 55 utterances for which the city name was not present in the 500-best list might benefit from adding the 'unigram+state' LM. This is the maximum gain possible.

The column denoted 'Unigram+state' in Table 1 shows the coverage results when using the utterance-dependent unigram+state LMs: There is an increase in coverage. Nine more correct rare city names (compared to the baseline set-up) are selected in the 500-best and the 3000-best lists, resulting in a coverage of 72.6% and 84.2%, respectively.

5.2. The second stage

The *N*-best lists generated by SpeM when using the unigram+state LMs to guide the search were included in the dynamic lexicon of the second stage recognizer. Subsequently, a new recognition was carried out. Table 3 shows the performance of the two-stage recognizer in terms of accuracy ('Acc. (%)') and number of rare city names that are correctly recognized ('#cities').

Table 3. Results of the two-stage recognizer for varying sizes of the *N*-best list generated by SpeM.

N	**0**	**500**	**1000**	**1500**	**2000**	**2500**	**3000**
Acc.(%)	65.5	75.6	75.9	76.0	76.2	76.2	76.4
#cities	0	87	87	87	88	88	88

What is immediately clear from Table 3 is that adding city names to the dynamic list increases the accuracy of the system by 10.1%, and 87 more city names are correctly recognized, while further adding city names does not increase the performance of the two-stage recognizer much. Analyzing the correctly recognized city names for the varying sizes of *N* revealed that 84 of the utterances correctly recognized for *N*=500, 1000, and 1500 are the same. For two utterances, the city name was correctly recognized when *N*=500, and not for *N*=1000 and 1500, and vice versa. For *N*>1500, one additional utterance was correctly recognized. Thus, although more of the correct city names are included in the *N*-best lists generated by SpeM; this does not result in an increase in performance. This matter is subject for further research.

In [5], it was also suggested that adding unigram scores to the city names in the dynamic list of the second stage recognizer might improve the performance of the second stage recognizer. To that end, we added the unigram scores to the city names in the dynamic list in the final experiment, and subsequently a new recognition was carried out. We used the 3000-best list; since it gave the best results (see Tables 1 and 3). The accuracy of the two-stage recognition system when using unigram scores in the second-stage recognizer was 75.9%, a decrease in accuracy of 0.5% absolute, but the number of correctly recognized city names increased with 2 to 90 (of the 203 city names found by SpeM (44.3%)).

6. Discussion and conclusions

In this work, we presented a two-stage recognition system for handling OOVs in a large vocabulary speech recognition task. We showed that SpeM, when using the 3000-best list, is able to retrieve over 84% of the rare city names that were missing from the first stage lexicon. Once the rare city names selected by SpeM were added to the lexicon of the second stage an increase in accuracy was obtained of 10.9% compared to the baseline in which no rare city names were added to the dynamic lexicon of the second stage.

The experiments presented in this paper showed that if no unigram counts are available, the two-stage recognition system still works reasonably well even though all words in the lexicon have equal probability. Adding state information when selecting the city names from the fallback lexicon, however, does improve the performance of the recognition system.

The eventual recognition results showed that just over 40% of the rare city names that were found by SpeM were correctly recognized. Analysis of the results showed that the city name that was not recognized correctly was often substituted by a city name with which it is highly confusable, e.g., 'Merryville' was substituted by Merrillville'. These errors can be tackled by improving the phone graph underlying the OOV intervals, and by improving the second stage recognizer.

7. Acknowledgments

The authors would like to thank Ed Filisko for providing the unigram counts and Louis ten Bosch and Lou Boves for fruitful discussions about this work.

8. References

[1] Seneff, S., "Response planning and generation in the Mercury flight reservation system", *Computer Speech and Language*, 16:283-312, 2002.

[2] Glass, J.R., Hazen, T.J., Hetherington, I.L., "Real-time telephone-based speech recognition in the Jupiter domain", *Proceedings of ICASSP*, Phoenix, AZ, p. 61-64, 1999.

[3] Geutner, P., Finke, M., Waibel, A., "Selection criteria for hypothesis driven lexical adaptation", *Proceedings of ICASSP*, Phoenix, AZ, p. 617-620, 1999.

[4] Bazzi, I., Glass, J.R., "Modeling out-of-vocabulary words for robust speech recognition", *Proceedings of ICSLP*, Beijing, China, p. 401-404, 2000.

[5] Scharenborg, O., Seneff, S., Boves, L., "A two-pass approach for handling OOVs in a large vocabulary recognition task", *Submitted to Computer Speech and Language.*

[6] Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., "How should a speech recognizer work?", *Accepted for publication in Cognitive Science.*

[7] Glass, J.R., "A probabilistic framework for segment-based speech recognition", *Computer Speech and Language*, 17:137-152, 2003.

[8] Chung, G., Wang, C., Seneff, S., Filisko, E., Tang, M., "Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation", *Proceedings of Interspeech 2004*, Jeju Island, Korea, p. 328-332, 2004.

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

[10] http://www.census.gov/main/www/cen2000.html

[11] Seneff, S., 'The use of subword linguistic modeling for multiple tasks in speech recognition", *Speech Communication*, 42(3-4):373-390, 2004.