Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 21 (2007) 206–218
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

A two-pass approach for handling out-of-vocabulary words in a large vocabulary recognition task

Odette Scharenborg [a,*], Stephanie Seneff [b], Lou Boves [a]

[a] *Centre for Language and Speech Technology, Radboud University Nijmegen, P.O. Box 9103, 6500 HD Nijmegen, The Netherlands*
[b] *Spoken Language Systems Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA*

Received 1 March 2005; received in revised form 2 March 2006; accepted 31 March 2006
Available online 27 April 2006

Abstract

This paper addresses the problem of recognizing a vocabulary of over 50,000 city names in a telephone access spoken dialogue system. We adopt a two-stage framework in which only major cities are represented in the first stage lexicon. We rely on an unknown word model encoded as a phone loop to detect OOV city names (referred to as 'rare city' names). We use SpeM, a tool that can extract words and word-initial cohorts from phone graphs from a large fallback lexicon, to provide an *N*-best list of promising city name hypotheses on the basis of the phone graph corresponding to the OOV. 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. It appeared that SpeM was able to include nearly 75% of the correct city names in an *N*-best hypothesis list of 3000 city names. With the names found by SpeM to extend the lexicon of the second stage recognizer, a word accuracy of 77.3% could be obtained. The best one-stage system yielded a word accuracy of 72.6%. The absolute number of correctly recognized rare city names almost doubled, from 62 for the best one-stage system to 102 for the best two-stage system. However, even the best two-stage system recognized only about one-third of the rare city names retrieved by SpeM. The paper discusses ways for improving the overall performance in the context of an application.

1. Introduction

The research described in this paper is part of a larger research project aiming at providing seamless domain switching among multiple domains in a single conversational agent. As a first step towards that goal, we have combined the vocabularies of two pre-existing systems, the Jupiter weather domain (Glass et al., 1999; Zue et al., 2000) and the Mercury flight domain (Seneff, 2002). There is a large overlap in the general lexicons of the Jupiter and Mercury systems. For instance, they share general question syntax and dates, as well as city names, making it a logical step to combine the two systems into one domain-independent system.

* Corresponding author. Tel.: +31 24 3611644; fax: +31 24 3612907.
E-mail address: O.Scharenborg@let.ru.nl (O. Scharenborg).

0885-2308/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2006.03.003

Jupiter is an on-line spoken dialogue system that provides weather forecasts via a toll-free telephone number. In its current configuration, Jupiter is able to handle requests for about 500 cities. Jupiter's weather source has recently been expanded such that it can now provide weather information for 38,000 cities, which means that all 38,000 city names need to be incorporated into the speech recognizer in some way. In this paper, the 500 city names originally included in the Jupiter lexicon are referred to as 'frequent' city names, while the newly added city names are referred to as 'rare city names'. A straightforward solution is to simply expand the recognizer's lexicon, which will, however, result in an extremely large search space, with only a back-off prior probability associated with each of the rare cities. Very large lexicons do not necessarily pose any (implementation) problem for automatic speech recognition (ASR) systems, but the combination with a weak language model usually results in poor performance.

To overcome the problem of a weak language model, we adopt here a novel strategy which uses small-sized lexicons in combination with a generic phone-based *unknown word* or *out-of-vocabulary* (OOV) *word* model to represent rare city names in the form of a phone sequence. This approach licenses in a second stage only those city names that match the proposed phone sequence sufficiently well (this will be explained in more detail in Section 2). The goal of this study is to build a two-stage recognizer that detects OOV in the first stage, and adapts the lexicon of the second stage on the basis of the phonemic composition of the OOV intervals, so that the rare city names can be recognized in the second stage.

In the literature, a variety of different solutions to handle OOV words have been proposed, which can roughly be divided into two groups. In the first group (exemplified by the Hypothesis Driven Lexical Adaptation (HDLA) method proposed by Geutner et al. (1999) and the Multi-pass Automatic Speech recognition uSIng Vocabulary Expansion (MASSIVE) method proposed by Ohtsuki et al. (2004)), a subset of words from a large fallback lexicon is selected on the basis of the results of the first stage recognizer. The selected subset is then added to the lexicon of the second stage recognizer. The second group of solutions omits a fallback lexicon, and thus other techniques have to be found to deal with the OOVs (e.g., decompounding strategies (Laureys et al., 2002); using a phone loop as an OOV 'word' parallel to the words in the lexicon (Bazzi and Glass, 2000, 2001)). In this research, a large fallback lexicon is available in the form of the list of city names (see Section 3). Therefore, in accordance with Geutner et al. (1999) and Ohtsuki et al. (2004), we built a two-stage recognizer that uses the outcome of the first recognition stage to create an adapted lexicon for the second stage recognizer by selecting a subset from the fallback lexicon.

To select the subset of words from the large fallback lexicon, the HDLA method (Geutner et al., 1999) uses morphology, phonetic, and grapheme distances, while MASSIVE (Ohtsuki et al., 2004) measures the distance between the input speech and the words in the vocabulary database in terms of word co-occurrence patterns. In this research, we use SpeM (SPEech-based Model of human speech recognition (Scharenborg et al., 2005)) – a tool originally designed for the simulation of human speech recognition (HSR) processes – to extract words and *word-initial cohorts* (words sharing phone prefixes) from the fallback lexicon on the basis of the similarity between the phones in a phone graph (produced by the first stage recognizer) and the phonemic representation of the words and word-initial cohorts in the fallback lexicon (see Section 2.2).

The selection of the correct rare city names from the fallback lexicon is of crucial importance for the performance of the second stage recognizer. Optimizing the coverage of the second stage lexicon is, therefore, the main focus of this work. In the first series of experiments (Section 4), we focus on the selection of the rare city names out of the fallback lexicon by SpeM. We investigate how often the correct rare city name is present in the subset for varying sizes of the subset. We investigate whether information about a recognized state name is helpful in increasing the number of correct rare city names in the subset. In the second series of experiments (Section 5), we investigate two methods to create a 'base' lexicon containing the 'general' words. The focus of these experiments is on the recognition performance of the second stage recognizer.

This paper is organized as follows. The following section describes the two-stage recognizer, its components, the way the recognition system detects and marks OOV words, and the general method SpeM uses to select the most likely words corresponding to the OOV intervals. Section 3 describes the materials used throughout this study. Section 4 focuses in more detail on the selection of the city names for the lexicon of the second stage. The results obtained with the two-stage recognition system on a speech recognition task are presented in Section 5, together with a discussion of the results. The paper ends with conclusions and suggestions for future research.

2. The proposed two-stage recognition system

The proposed two-stage recognition system is schematically depicted in Fig. 1. The acoustic signal is fed into the first stage recognizer, which uses a lexicon that captures 'general' words (see Section 3 for more details) in addition to the 500 most frequent city names (originally from the Jupiter system). Since the method we propose to deal with OOV words in a two-stage recognition system is crucially dependent on the detection of the OOV intervals by the first stage recognizer, an OOV model that is intended to mark all city names not in the lexicon as being OOV is integrated into the first stage (see Section 2.1.3). The hypothesized phone graphs corresponding to 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 for that specific utterance from the fallback lexicon. This subset of most likely rare city names is then added to the 'utterance-dependent' lexicon of the second stage (see Section 4). The second stage recognizer then does a new recognition on the basis of the same acoustic models as were used in the first stage (see Section 5). In the second stage, the lexicon (and thus the recognizer) is tuned to the utterance (and thus the domain).

2.1. Automatic speech recognition system

The two-stage recognizer used in this study is the segment-based automatic speech recognition system SUMMIT (Glass, 2003), which uses finite state transducers (FSTs) to represent its search space. The entire linguistic search space in the recognizer can be represented by a single FST U, which is represented by a cascade of FST compositions:

$$U = C \circ P \circ L \circ G \tag{1}$$

where C contains the mapping from the diphone labels used in the acoustic model set to monophone labels, P applies phonological rules, L maps the words in the lexicon to their phonemic representations and G is the language model (LM).

2.1.1. The language models

The two-stage recognizer uses class bigram and trigram LMs. For the class n-gram LMs, we defined 70 classes, for example, for months, weekdays, airlines, and weather related adjectives. But most importantly for this research are the definitions of two city name related classes: one containing the frequent city names and one containing the rare city names. Subsequently, the class bigram and trigram LMs were trained on 167,967 utterances (on average 5.8 words per utterance) from both the Jupiter and the Mercury domain. The number of

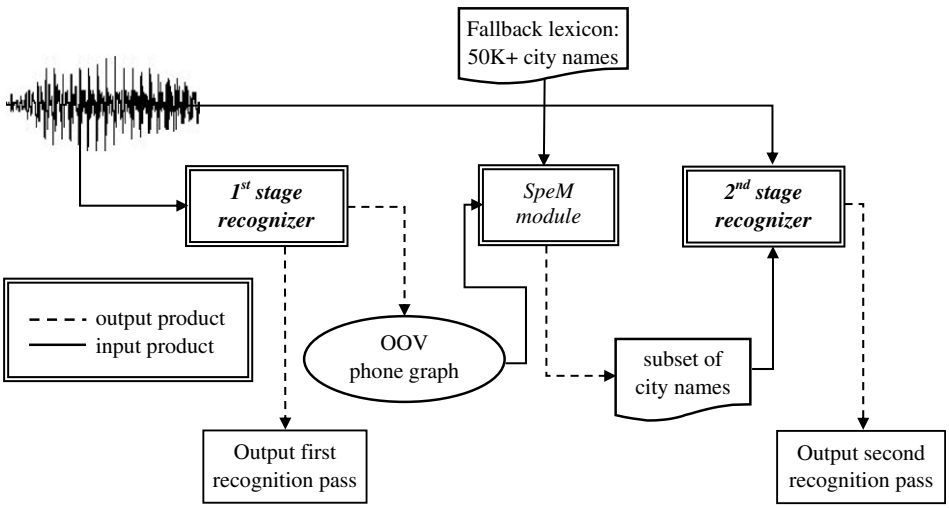

Fig. 1. Overview of the proposed multi-stage recognition system.

rare city names in the training material was rather low; this would result in a very small probability for the rare city class. To increase the number of rare city names and thus the probability for entering the rare city class, 3000 frequent city names in existing utterances were randomly replaced by a rare city name, and were thus treated as a rare city name. In this way, frequent and rare city names were treated as different classes and separate LM scores were calculated for both classes. In addition to the probability for entering the frequent city name or rare city name class, the frequent city names each have a unigram score. Since no unigram scores are available for the rare city names, all city names in the rare city name class have equal probability ($-\log n$, where n is the number of rare city names included in the rare city name class). For more details about the class LMs, see Seneff et al. (2003).

For the calculation of the perplexity of the bigram and trigram LMs, all rare city names were mapped onto a single 'word', while all frequent city names (like all other words) were treated as distinct words. For each sentence, an FST was constructed for the word string. Since a word can belong to several classes, e.g., 'one' is both a numeral and a pronoun, the FST was subsequently composed with a 'mapping' FST that adds all possible 'parses' for words. For each parse, the best n-gram score (in terms of negative log probabilities) is calculated. The perplexity is then defined as follows:

$$\text{Perplexity} = e^{\text{total\_log prob}/\text{total\_words}}, \tag{2}$$

where total_log prob is the sum of all negative log probabilities and total_words is the total number of words. The perplexity of the trigram LM was 13.1, the perplexity of the bigram LM was 16.1. Clearly, mapping all rare city names onto one single 'word' has a big effect on keeping perplexity small.

Both stages of the recognition system use the same bigram and trigram LMs. This implies that the LMs of the second stage recognizer are not tuned to the utterance as is the case for the lexicons. Nevertheless, our approach alleviates the problem outlined in the introduction: although the rare city names still have a relatively small prior probability, the probability of entering the *class* of the rare city names is much higher; thus the probability of recognizing the existence of a rare city name will increase. Furthermore, the unigram probability within the class of rare city names decreases with an increasing number of words in the lexicon. Since only a subset of the full fallback lexicon is added to the lexicon of the second stage, the unigram score for each rare city name is higher than it would have been if all rare city names were added to the lexicon.

2.1.2. The acoustic model set

SUMMIT uses a segment-based framework to create an acoustic–phonetic representation of the speech signal in the form of a phonetic segment network (Glass, 2003). The basis for this network (or graph) is probabilistic–acoustic landmarks, which correspond to either a boundary between two segments or a segment-internal event (which indicates a significant acoustic event within a segment). The segments are phonetic units. Feature vectors are extracted both over hypothesized phonetic segments and at their boundaries for phonetic analysis.

In total, 1389 distinct acoustic models were trained on 20,296,714 training tokens to model the boundaries and the segments. Each acoustic model has on average 30.08 Gaussians (with a maximum of 100 Gaussians), resulting in a total number of Gaussians for the complete system of 41,777. The feature vectors are based on Mel-scale Cepstral coefficients.

2.1.3. Detecting the OOVs

The procedure used to mark the OOV words and generate the OOV phone graphs is described in detail in Bazzi and Glass (2000, 2001): the *generic word model* is implemented as a phone loop that allows for arbitrary phone sequences during recognition. This OOV model is included in the lexicon (L) and as such wired into the linguistic search space (U). 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. The OOV penalty was tuned on a development set such that the number of OOVs marked by the first stage recognizer was in line with the number of actual OOVs in the development set, and the recognition accuracy did not deteriorate substantially. The development set contained 2914 utterances from both the Mercury and the Jupiter domains; not all of the utterances contained an OOV. The value of the OOV penalty was the same in both the first and the second stage recognizer, since it is possible that OOVs occur in the second stage recognizer.

Underlying the hypothesized OOV is the OOV phone graph. For each utterance in which an OOV was hypothesized in the word lattice, only one OOV phone graph was generated (this is due to the implementation of the procedure to extract the OOV phone graphs). This means that, where an utterance contains more than one OOV, as in *I'd like to fly from* ⟨OOV⟩ *to* ⟨OOV⟩, an underlying OOV phone graph can be generated for only one of the OOVs. In the test data used in this study, however, each utterance contained at most one rare city name.

Note that an OOV might be hypothesized for a stretch of the speech signal that does not correspond to a rare city name. Furthermore, it is possible that the phone graph does not match exactly with the stretch of speech that contains the rare city; i.e., it is possible that additional phones are included at the start or the end of the phone graph that do not belong to the rare city name. Likewise, it is possible that the OOV word is truncated, i.e., phones of the rare city name are missing at the start or the end of the phone graph. Finally, it is possible that the first stage recognizer incorrectly recognizes the rare city name as an in-vocabulary word.

2.1.4. The 'dynamic' lexicon

The recognizers in the first and second stage are identical, with the exception of the lexicon (L). The lexicons of the first and second stage consist of three components, as shown in Fig. 2. The first two components, the 'Base' lexicon and the OOV 'word' model are similar for both recognizers. The difference between the two lexicons is in the 'dynamic' lexicon. The dynamic lexicon can be wired on-the-fly into the search space of the recognizer, without the need to rebuild the lexical search space (Chung et al., 2004b). For the first stage recognizer, this dynamic lexicon is empty; for the second stage recognizer, this dynamic lexicon is supplied with the list of rare city names extracted by SpeM from the fallback lexicon (see also Sections 2.2 and 3).

2.2. SpeM

SpeM (Scharenborg et al., 2005) was originally implemented to serve as a tool for research in the field of human speech recognition. It is a new and extended implementation of the theory underlying the *Shortlist* model, a computational model of human word recognition developed by Norris (1994). The main advance of SpeM over pre-existing computational models of human word recognition is that SpeM uses the acoustic speech signal as input, while Shortlist and other computational models of HSR only take handcrafted symbolic representations (e.g., phonemes or phonetic features) as input. Furthermore, SpeM supports unigram and bigram language models (see Section 2.2.2), unlike Shortlist and the other HSR models. However, in the experiments described in this paper, only a unigram LM is used. Besides its use as a tool for simulating results found in HSR experiments, SpeM can also function as an experimental ASR system, e.g., for the recognition of words before the acoustic realization of the word is complete (Scharenborg et al., in press).

SpeM consists of two modules: an *automatic phone recognizer* (APR) and a *word search module* (see Section 2.2.1). The word search module parses the probabilistic phone graph created by the APR in order to find the most likely (sequence of) words (Scharenborg et al., 2005). In the experiments described in this paper, the phone graphs are created by the first stage recognizer. In the remainder of this paper 'SpeM' only refers to the word search module.

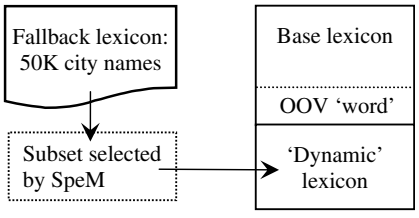

Fig. 2. The components of the lexicons of the first and second stage recognizer.

2.2.1. The word search

In SpeM, the sequence of words with the smallest distance between the sequence of phones on a path through the OOV phone graph and the phonemic representations of the words in the fallback lexicon (represented as a lexical tree) is determined using a time-synchronous and breadth-first dynamic programming (DP) algorithm.

Each phone insertion, deletion, and substitution is penalized according to their own penalty which can be tuned separately (for more details the reader is referred to Scharenborg et al. (2005)). Furthermore, a garbage phone model is included in the lexicon. This garbage phone model is mapped onto phones appearing at the start and at the end of the phone graph that belong to the preceding and following word. Spurious phonemes preceding or following a rare city name in the OOV phone graph are mapped onto these garbage phones. Missing phonemes, on the other hand, are treated as deletions by SpeM.

The output of SpeM consists of an *N*-best list of hypothesized parses. Each parse contains words, word-initial cohorts, garbage, silence, and any combination of these, with the exception 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. Although SpeM is able to recognize sequences of words, internal parameters were set such that the recognition of a single word (preceded or followed by garbage) is more likely.

Subsequently, if a word-initial cohort has been recognized for an utterance (or OOV phone graph), and if it consists of more than three phones, the word-initial cohort is 'unpacked': all words belonging to the word-initial cohort are listed. Finally, for each utterance (or OOV phone graph), the top *N* words at the output of SpeM are selected into the utterance-specific *N*-best list that goes into the utterance-specific dynamic lexicon of the second stage recognizer. The effect of the size of these *N*-best lists is investigated in Section 4.

Note that SpeM always returns an *N*-best list of most likely rare city names, even if the phone graph did not correspond to a city name. This is because SpeM searches a lexicon that contains only city names.

2.2.2. The language models in SpeM

SpeM is able to use unigram (and bigram) scores during its word search process. In that case, the ranking of the hypotheses in the resulting *N*-best list are not only based on the acoustic score (see previous section), but also on the language model scores. For example, if there are two competing hypotheses with equal acoustic scores, the hypothesis with the larger language model score(s) will be ranked higher than its competitor.

3. Materials

The experiments were conducted on a set of continuous speech utterances, recorded from interactive telephone conversations with both the Mercury and the Jupiter systems. The independent test set consisted of 418 utterances taken from both domains, each utterance containing exactly one rare city name. The first stage recognizer did not detect an OOV in 19 utterances of the test set (4.5%), which means that the rare city name was recognized incorrectly as an in-vocabulary word. The utterances for which no OOV was detected were discarded from the test set – since no improvement of the recognition of the rare city name by the second stage recognizer can be expected – leaving 399 utterances that were used in the experiments.

The lexicon of the first stage consisted of the 'general' words from both domains, a short list of the 500 most frequent city names, all US state names, and a set of 1326 partial and short city names with a phonemic representation of three phones or less, such as 'los', 'ann', 'new' – this to simplify SpeM's task, since short words are difficult to find in a phone lattice. This resulted in a lexicon of 2802 words. Note that other OOV words besides rare city names can occur. In our complete test set, 10 words (in nine utterances) other than rare city names were missing from the first stage recognizer's lexicon. In only one of those nine utterances, the phone graph that was extracted did not correspond to the phone graph underlying the rare city name. In that case, SpeM is of course unable to find the correct rare city name. Thus, the first stage recognizer was able to tag 398 of 418 rare city names (95.7%) as OOV.

The fallback lexicon is not limited to the 38,000 city names Jupiter knows since, on the one hand, we want to cover not only those city names for which Jupiter has the weather information, but, more generally, any city name that the user might utter (both for flight and weather information) and, on the other hand, a user cannot precisely know which 38,000 city names Jupiter knows. The fallback city name lexicon, therefore, contains

52,595 city names, which were harvested from the Geographic Names Information System website (GNIS: http://geonames.usgs.gov/stategaz/). The city names were extracted from the 'city column' from the database. The number of typing errors was small, and they were changed whenever one was discovered. All of the rare city names observed in the 399 test utterances occurred in the fallback lexicon. The 38,000 city names Jupiter knows and the 500 most frequent city names (which are also present in the first stage recognizer's lexicon) are a subset of the fallback lexicon.

Most of the city names were non-existent in our lexical baseforms resource file, and pronunciations were therefore automatically generated for them using the letter-to-sound system[1] described in Chung et al. (2004a) and Seneff (2004). The errors in these pronunciations have not been corrected manually. This further challenges the recognition task.

4. Extracting the subset from the fallback lexicon

In this section, we focus on the performance of SpeM in selecting the correct city names from the fallback lexicon into the utterance-dependent *N*-best lists. Two factors are investigated: (1) Does recall increase proportionally with the increase of the size of the *N*-best list? Or, does SpeM behave similarly to standard beam search techniques, in which a small beam width is already able to maintain the hypotheses that are most promising and to suppress the hypotheses that are unlikely, such that broadening the beam width does not improve performance much, or even hurts recognition performance (Ney and Ortmanns, 2000)? (2) Can recall be improved by using state names present in the output of the first stage recognizer to build utterance-dependent unigram language models in SpeM?

The results of this experiment are presented in terms of coverage, i.e., the percentage of the test set utterances for which the rare city name in its transcription (which was presumably marked as OOV by the first stage recognizer) is present in the *N*-best list generated for that utterance by SpeM.

SpeMs performance in selecting the correct rare city names from the fallback lexicon is not only dependent on the search algorithm of SpeM, but also on the quality of the OOV phone graphs. To be able to assess SpeM's performance, we, therefore, first determined the quality of the OOV phone graphs.

4.1. The quality of the OOV phone graphs

We investigated the quality of the OOV phone graphs by analyzing the recognition results of the first stage recognizer. We calculated how often the word immediately preceding and immediately following the hypothesized OOV were correctly recognized. This does not give an exact measure of the quality of the OOV phone graph, but it does provide insights as to whether phonemes belonging to the words immediately preceding and following the rare city name are included in the OOV phone graph or whether the OOV phone graph has too few phonemes. The results showed that 14.5% of the words preceding and 21.1% of the words following the OOV were not correctly recognized. In these cases, it is to be assumed that the time alignment of the underlying OOV phone graph with the rare city name is not completely accurate. These results indicate that, in the worst case, 35.6% of the phone graphs are defective. However, the actual proportion is somewhat lower, because there are utterances in which both the preceding and following word are incorrect. Although in the majority of the cases the words immediately preceding and following the rare city name are correctly recognized, it is evident that there is room for improvement in detecting the OOV stretch.

4.2. Recall as a function of the length of the N-best list

The size of the utterance-dependent *N*-best lists created by SpeM was varied between 500 and 3000 in steps of 500 entries. The results are shown in Table 1. The second column (denoted 'Utt-indep LM') of Table 1 shows

[1] The performance of the letter-to-sound system was not evaluated for the task at hand. Earlier results by Chung et al. (2004a) on 198 isolated words drawn from the OGI Names Corpus (Names 1.3, the CSLU OGI names corpus: http://cslu.cse.ogi.edu/corpora/name/) showed that 30.8% of the names had one or more erroneous phonemes. Unfortunately, phone accuracy information for the names task is not available.

Table 1
OOV-recall results and analysis for varying sizes of the SpeM N-best lists

N-best list size	Utt-indep LM		Max. Gain	Utt-dep LM		Analysis	
	Abs	%		Abs	%	Loss	Gain
500	223	55.9	102	275	68.9	6	58
1000	230	57.6	98	281	70.4	7	58
1500	234	58.6	95	284	71.2	9	59
2000	235	58.9	95	291	72.9	9	65
2500	238	59.6	93	293	73.4	11	66
3000	239	59.9	92	296	74.2	10	67

recall for the varying sizes of the N-best lists in terms of absolute number of utterances for which the correct rare city name was present in the N-best list ('Abs') and as a percentage of the 399 utterances of the test set ('%').

The recall results show that already over 55% of the rare city names that were missing from the lexicon of the first stage recognizer are now present in the lexicon of the second stage. This is an encouraging result, bearing in mind that all 52,595 words in the fallback lexicon have equal probability, and that some 30% of the generated OOV graphs are not perfectly segmented. Comparing recall for the N-best sizes 500 and 3000 clearly shows that increasing the length of the N-best list 6-fold does not increase recall with the same amount: only 16 more correct rare city names were found when the N-best list size was 3000. This shows that the lexical search of SpeM works in a way comparable to the beam search used in standard ASR systems: a relatively small beam width is already able to preserve the most promising hypotheses while suppressing the most unlikely.

4.3. Utterance-dependent language model

It might be possible to improve on the results shown in the 'Utt-indep LM' column in Table 1 if city names that are more likely on the basis of the context of the utterance receive a higher prior probability. An obvious cue is the state name. It is highly likely that a city name, which is uttered in the same utterance as a state name, lies in that state. Thus, if a state name has been recognized by the first stage recognizer, that information can be used to increase the prior probability of all city names in that state. The database with state-city name information that we have available shows that there are on average 1890 city names per state. So, if the state name were known, it would reduce the number of possible city names considerably.

In this experiment, we built utterance-dependent unigram language models[2] for SpeM for those utterances in which a state name was recognized. If a state name is present in the N-best list[3] (in our experiments, $N = 50$) generated by the first stage recognizer, all city names in that state receive a higher unigram score. If an N-best list contains more than one state name, all city names in all those states receive a boost. The optimal boost was determined in a series of tuning experiments.

In our test set, for 299 of 399 utterances (74.9%) a state name is present. For 243 utterances, a state name appeared in the N-best lists generated by the first stage recognizer. For these utterances, utterance-dependent language models were created. Of course, the 56 utterances in which a state name was present but not found by the first stage recognizer and the 100 utterances in which no state name was present will not benefit from this approach.

First, we tabulated for how many of the 243 utterances for which a state name was recognized by the first stage recognizer the correct rare city name was already present in the N-best list generated by SpeM. In this way, the maximum gain could be calculated. The column 'Max. Gain' in Table 1 contains the number of utterances in which the first pass recognizer found a state name, and in which SpeM could not find the correct city name in the first experiment.

The fourth column denoted 'Utt-dep LM' shows recall in terms of absolute number of utterances ('Abs') and the percentage of the full test set ('%') when the utterance-dependent language models were added to SpeM. As can be seen from this column, there is a clear increase in recall. In the case of an N-best list of

[2] Note that these language models are not related to the language models used in the first and second stage recognizer.
[3] Note that these N-best lists are independent from the N-best lists generated by SpeM.

500, the correct rare city name was selected into the utterance-dependent lexicon for 52 more utterances; for an *N*-best list of 3000, this number is slightly larger: 57, resulting in a recall of 74.2%.

4.4. Analysis and discussion

There is always the risk that the state name recognized by the first stage recognizer is incorrect, or that another word in the utterance is incorrectly recognized as a state name, which will result in a boost of the probability of the wrong city names. Therefore, we also analyzed the number of utterances for which the utterance-dependent language models made the correct rare city disappear from the *N*-best list by SpeM. These results are shown in the column 'Analysis' in Table 1. For instance, in the case of an *N*-best list of length 500, for six utterances for which the correct rare city had been selected into the *N*-best list with the utterance-independent LM, the correct rare city name was no longer selected with the utterance-dependent LMs. On the other hand, for 59 utterances for which the correct city name was missing from the *N*-best lists generated by SpeM, the correct rare city names were selected once the utterance-dependent LMs were used. The risk that a rare city name, which was originally in the *N*-best list, drops out with the utterance-dependent language models increases with increasing size of the *N*-best list: for the 500-best list, the 'Loss-Gain' ratio was 10.3 (6/58), while for the 3000-list it was 14.9 (10/67). The longer *N*-best lists contain more rare city names than the shorter *N*-best lists, but the acoustic scores, on which the ranking within the *N*-best list is based, are relatively small for a lot of rare city names in the longer *N*-best lists. When unigram scores are added, competitor city names with an equal or even lower acoustic score might exceed the score of the city name causing the latter to drop from the *N*-best list. However, one might wonder whether the second stage recognizer would have been able to correctly recognize these rare city names, since apparently their acoustic match is rather small.

Comparing the 'Gain' and 'Max. Gain' columns shows that there is room for improvement: adding the utterance-dependent LMs increases the number of correct rare city names in the *N*-best lists, but still not all rare city names are found by SpeM. Additional methods need to be found to further improve on the selection of the correct rare city name.

In conclusion, using the utterance-dependent language models resulted in a net gain in recall: 68.9% to 74.2% (depending on the size of the *N*-best list created by SpeM) of the rare city names that were missing from the first stage lexicon are now included in the lexicon of the second stage recognizer. This will increase the probability that the second stage recognizer ultimately recognizes the correct rare city name. In the recognition experiments discussed in the next section, we always used the *N*-best list generated by SpeM with the utterance-dependent language models.

5. Performance of the two-stage recognizer

For each of the utterances in the test set, an utterance-dependent *N*-best list of rare city names that can be added to the dynamic lexicon of the second stage recognizer is now available. The full system, with the base lexicon and the *N*-best lists generated by SpeM, now can be used to run a full recognition for each utterance of the test set. What remains to test our multi-stage recognizer is the construction of the base lexicon for the second stage recognizer (see Fig. 2). We investigated two methods to construct the base lexicon: one using the same base vocabulary as in the first stage, and the other using a much smaller lexicon derived directly from the output of the first stage recognizer.

To asses the performance of our multi-stage recognizer, first a baseline experiment is carried out in which all the city names of the fallback lexicon are included in the *dynamic* lexicon of the first stage recognizer – which was previously empty. Theoretically, no OOV city names exist. The performance of the recognition systems are measured in terms of accuracy, but since we are mainly interested in the recognition of the rare city names, the number of correctly recognized rare city names is also counted and presented.

5.1. Baseline experiment: adding all city names to the lexicon

In the baseline experiment, all 52,595 city names from the fallback lexicon are available in the first stage recognizer. In analogy with our two-pass approach, all city names from the fallback lexicon are added to

the dynamic lexicon of the first stage recognizer. The class bigram and trigram LMs used in the baseline experiment are identical to the class bigram and trigram LMs used in the multi-stage system. The unigram probability for a city name in the dynamic list is set to $-\log n$ (where n is the number of rare city names in the dynamic lexicon viz. 52,595).

Testing the one-pass system on the 399 test utterances showed a word accuracy of 72.6%. Of the 399 rare city names, 56 were correctly recognized (14.0%), while 36 times a rare city name was incorrectly hypothesized. Furthermore, 90 times an OOV was hypothesized. So, even though the rare city names were in the lexicon, OOVs were still being hypothesized and only 56 rare city names were correctly recognized.

One could argue that since all rare city names are explicitly known in the first stage recognizer, an OOV model is no longer necessary. We, therefore, adapted the OOV penalty such that OOVs could no longer be hypothesized, and tested the one-pass system ('without-OOV system') on the same test utterances. The results showed a word accuracy of 70.6%, while 62 rare city names were correctly recognized (15.5%). Thus, eliminating the option of hypothesizing OOVs increased the number of correctly recognized rare city names, but reduced the word accuracy compared to the one-pass system which was able to hypothesize OOVs ('with-OOV system').

The reduction in word accuracy for the without-OOV system might seem counterintuitive, but is easy to explain. Eliminating the possibility to hypothesize an OOV is likely to hurt the recognition of the surrounding words as well: the without-OOV system showed a substantial increase in number of insertions (mostly of short words) compared to the with-OOV system. Furthermore, the small increase in the number of correctly recognized rare city names can be explained by the fact that only a few rare city names benefit from the elimination of the OOV model, because the hypothesis of an OOV in the with-OOV system means that the likelihood of the rare city name was already rather low. The without-OOV system now hypothesizes an incorrect city name for them.

The rather low performance with respect to the number of correctly recognized rare city names for both one-pass systems is due to confusability within the dynamic lexicon, on the one hand, and the low unigram probability of the city names within the dynamic lexicon, on the other.

5.2. The Ohtsuki method for creating the second stage base lexicon

The easiest way to construct the base lexicon of the second stage recognizer is to use the same lexicon as was incorporated into the first stage recognizer. The dynamic lexicon is then in effect an addition to the first stage recognizer's lexicon. This method has also been used by Ohtsuki et al. (2004) for a similar task.

The column denoted 'Ohtsuki method' in Table 2 presents the results for varying sizes of the N-best lists generated by SpeM. The size of '0' is the baseline result of the Ohtsuki method: in this case, only the base lexicon (including the OOV model) is used for the recognition. The baseline system is thus identical to the first stage recognizer. Again, the unigram probability for a city name in the dynamic list is set to $-\log n$ (where n is the number of rare city names in the dynamic lexicon, viz. between 500 and 3000), but since the number of words in the dynamic lexicon is much lower than in the case of the one-stage recognizer, the actual unigram probability is much higher.

Table 2
Results of the two-stage recognizer for varying sizes of the N-best list generated by SpeM, and two different methods to construct the base lexicon

N-best list size	Ohtsuki method			Tang method		
	Accuracy (%)	#Rare cities	Lex. size	Accuracy (%)	#Rare cities	Lex. size
0	68.3	0	2802 + 0	69.3	0	±23.5 + 100 + 0
500	77.3	101	2802 + 500	76.9	106	±23.5 + 100 + 500
1000	77.0	97	2802 + 1000	77.3	104	±23.5 + 100 + 1000
1500	76.8	96	2802 + 1500	77.1	101	±23.5 + 100 + 1500
2000	76.7	95	2802 + 2000	77.1	101	±23.5 + 100 + 2000
2500	76.7	93	2802 + 2500	76.9	100	±23.5 + 100 + 2500
3000	76.4	93	2802 + 3000	76.8	100	±23.5 + 100 + 3000

As shown in Table 2, the system with an N-best list size of 0 has an accuracy of 68.3%; adding an N-best list with the 500 most likely city names proposed by SpeM already increases the accuracy by 9.0% points (significant at the 0.95 level), while 101 rare city names are recognized correctly. Further increasing the lexicon size, however, improves neither the accuracy (no significant differences at the 0.95 level) nor the number of correctly recognized rare city names. The latter is most likely due to the similarity of the words in the N-best lists generated by SpeM. The words are similar because they are the most likely words from the fallback lexicon given the phone graph corresponding to the OOV stretch. This increases the confusability of the words in the lexicon, which in turn puts a curb on the maximum accuracy that can be obtained.

5.3. The Tang method for creating the second stage base lexicon

Tang et al. (2003) demonstrated that, for a two-stage recognition system, the most important words to retain in the lexicon of the second stage are the in-vocabulary words that have been hypothesized by the first stage recognizer, augmented with a list of those words that are most often deleted by the first stage recognizer. These words are usually short function words such as 'a', 'the', 'to', etc. Although the two-stage recognition system designed by Tang et al. was used for the recognition of the sub-word phonetic features 'manner' and 'place of articulation', we think that Tang's method might be useful to decrease the size of the second stage lexicon in our research.

The base lexicon in the Tang method consisted of the 100 words that were most often deleted by the recognizer in the first stage, plus all words in the 50-best list created by the recognizer in the first stage (about 23.5 words per utterance on average). Again, the unigram probability for a city name in the dynamic list is set to $-\log n$.

The results are presented in the 'Tang method' column in Table 2. The lexicon size is reduced dramatically (compare the two columns denoted 'Lex. size'). The baseline system (N-best list size $= 0$) shows a higher accuracy than the baseline system when the base lexicon was made following the Ohtsuki method. This indicates that the internal confusability within the base lexicon has decreased, due to the decrease in the size of the base lexicon.

For the Tang method, the best accuracy is obtained when 1000 of the most likely words are added to the dynamic list (this result is significantly better than the baseline ($N = 0$)). As can be seen in Table 2, this accuracy is equal to the best accuracy obtained with the Ohtsuki method. However, the number of correctly recognized rare city names when using the Tang method is higher than with the Ohtsuki method (but not significantly). The highest number of correctly recognized rare city names, however, is obtained for an N-best list size of 500 with the Tang method. In this case, 106 of the rare city names have been recognized correctly, as contrasted with only 101 for the equivalent N-best size using the Ohtsuki method.

5.4. Analysis and discussion

The results in Table 2 show that an N-best list of size 500 is more favorable than a longer N-best list. This outcome is further backed-up by the fact that the results in Table 1 suggest that longer N-best lists contain more rare city names with a low acoustic score, because the names in the tail of the lists drop out when state name information is used in the SpeM language models.

We further analyzed the results of the two methods by calculating the 'accuracy' of the second stage recognizer in terms of the percentage of rare city names that were detected by SpeM (see Table 1, column 'Utt-dep LM') and which were subsequently correctly recognized by the second stage recognizer (see Table 2, columns '#rare cities'). The results are presented in Table 3. The values in the columns denoted 'Accuracy (%)' are derived by dividing the numbers of correctly recognized rare city names in Table 2 by the corresponding number of rare city names retrieved by SpeM in Table 1 and multiply by 100%. These figures clearly show that although SpeM is able to retrieve many of the rare city names from the fallback lexicon, less than 40% of these rare city names are subsequently recognized correctly by the second stage recognizer.

Although there is still much to gain in improving the second stage recognizer, comparing these results with our baseline experiments clearly shows that our proposed two-pass approach outperforms the (without-OOV and with-OOV) one-pass systems. Both the differences in accuracy (72.6%/70.6% vs. 77.3%) and the differences in number of correctly recognized rare city names (56/62 vs. 101/106) are significant at the 0.95 level. The two

Table 3
The accuracy (i.e., the percentage of the rare city names present in the dynamic lexicon that were correctly recognized by the two-stage recognizer) for varying sizes of the *N*-best list generated by SpeM, and the two different methods to construct the base lexicon

N-best list size	Accuracy (%)	
	Ohtsuki method	Tang method
0	0	0
500	36.7	38.5
1000	34.5	37.0
1500	33.8	35.6
2000	32.4	34.7
2500	31.7	34.1
3000	31.4	33.8

two-pass systems obviously benefit from the reduced size of the rare city name lexicon and the resulting higher unigram probabilities. Since the unigram probability for a city name in the dynamic lexicon is set to $-\log n$, a smaller size of the rare city name lexicon (as is the case for the two-pass systems) results in a higher unigram score for each rare city name than it would have been if all rare city names were added to the lexicon (as is the case for the one-pass systems), since n is much smaller.

The fact that less than 40% of the rare city names that were present in the lexicon of the second stage recognizer were indeed recognized correctly may seem surprising. However, it must be realized that SpeM inevitably selects a highly confusable subset from the fallback lexicon, which is a difficult challenge for the acoustic models. In addition, it is to be expected that some 30% of the entries in the fallback lexicon contain one or more errors in their phonemic representation (cf. footnote 1). A manual clean-up of the lexicon might already help to improve the results significantly, both in terms of the recall in SpeM and the accuracy of the second stage recognizer. Last but not least, since the prior probability of all names entered into the lexicon is equal, the recognizer can only rely on the acoustic match to make a decision. In an operational application one might add estimates of the prior probabilities based on population statistics, possibly combined with additional information, like the number of visitors in ski and beach resorts. A similar approach has appeared to be successful in a Directory Assistance application (Bouwman and Boves, 2001).

The ease of use of Galaxy, Jupiter, and Mercury with a largely extended list of city names can further be enhanced by means of a clever interface design. If a city name proposed by the system is not confirmed by the user, the system could first ask for (a confirmation of) the state name, then ask to speak or type the first letter of the city name and re-run the second stage recognizer for the recording of the erroneously recognized input in combination with a lexicon limited to the city names in the correct state that begin with the correct first letter. As shown in Sturm and Boves (2005) this procedure returns the correct city name in a large majority of the cases.

6. Conclusion and future work

In this work, we presented a two-stage recognition system for handling OOVs in a large vocabulary speech recognition task. We showed that SpeM is able to retrieve nearly 75% of the rare city names from a large fallback lexicon. Adding these rare city names to the lexicon of the second stage recognizer resulted in a recognition accuracy of the two-stage recognition system of 77.3%, an increase of 8.0% and 9.0% respectively for the two types of base lexicons we used, while the best one-stage system yielded a word accuracy of 72.6%. Even more importantly, the number of correctly recognized rare city names almost doubled, from 56 or 62 for the one-stage systems to 106 for the best two-stage system. These results are remarkable, keeping in mind that all words in the dynamic lexicon have equal prior probability.

The fact that SpeM was able to find the correct city name given a phone graph in nearly 75% of the utterances is encouraging. However, it is evident that there is substantial room for improvement. One way to do this would be using population statistics to estimate unigram probabilities for the city names. Improved performance might also be obtained by training phone-specific substitution, deletion, and insertion costs. Finally, future research should shed light on the impact of recognition errors in the words preceding and following the rare city name on the contents of the OOV phone graph.

The recognition results of the second stage recognizer showed that only about 1/3 of the rare city names that were found by SpeM were correctly recognized. This suggests that future work should focus on improving the second stage performance, rather than on improving the recall of rare city names in SpeM. Here too, the unigram probabilities of the words in the dynamic list could be improved by using population statistics. Secondly, the performance of the second stage recognizer might improve when an utterance-dependent language model is used. A first experiment would involve running parallel language models for the two domains – Mercury and Jupiter – in the second stage, and weighting them according to the likelihood of each domain, estimated from the word sequences extracted from the first stage.

Acknowledgements

Part of this work was carried out while the first author was visiting the Spoken Language Systems Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. The first author would like to express her appreciation for the hospitality of MIT and the scholarship (from the Katrien van Munster fund) awarded by the Radboud University Nijmegen; without both the visit would not have been possible. Furthermore, the authors would like to thank Lee Hetherington for his help in building the recognition system, providing the information about the acoustic models, and the calculation of the perplexity of the language models. Finally, the authors would like to thank Chao Wang for her help building the recognition system, Ed Filisko for providing the city names lexicon, and Min Tang for providing the set of test utterances.

References

Bazzi, I., Glass, J.R., 2000. Modeling out-of-vocabulary words for robust speech recognition. In: Proceedings of ICSLP, Beijing, China, pp. 401–404.

Bazzi, I., Glass, J.R., 2001. Learning units for domain-independent out-of-vocabulary word modeling. In: Proceedings of Eurospeech, Aalborg, Denmark, pp. 61–64.

Bouwman, G., Boves, L., 2001. Using information on lexical stress for utterance verification. In: Proceedings of ITRW on Prosody in ASRU, Red Bank, NJ, pp. 29–34.

Chung, G., Wang, C., Seneff, S., Filisko, E., Tang, M., 2004a. Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation. In: Proceedings of Interspeech, Jeju Island, Korea, pp. 328–332.

Chung, G., Seneff, S., Wang, C., Hetherington, I.L., 2004b. A dynamic vocabulary spoken dialogue interface. In: Proceedings of ICSLP, Jeju Island, Korea, pp. 327–330.

Geutner, P., Finke, M., Waibel, A., 1999. Selection criteria for hypothesis driven lexical adaptation. In: Proceedings of ICASSP, Phoenix, AZ, pp. 617–620.

Glass, J.R., 2003. A probabilistic framework for segment-based speech recognition. Computer Speech and Language 17, 137–152.

Glass, J.R., Hazen, T.J., Hetherington, I.L., 1999. Real-time telephone-based speech recognition in the Jupiter domain. In: Proceedings of ICASSP, Phoenix, AZ, pp. 61–64.

Laureys, T., Vandeghinste, V., Duchateau, J., 2002. A hybrid approach to compounds in LVCSR. In: Proceedings of ICSLP, Denver, CO, pp. 697–700.

Ney, H., Ortmanns, S., 2000. Progress in dynamic programming search for LVCSR. Proceedings of the IEEE 88 (8), 1224–1240.

Norris, D., 1994. Shortlist: A connectionist model of continuous speech recognition. Cognition 52, 189–234.

Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., 2005. How should a speech recognizer work? Cognitive Science 29 (6), 867–918.

Scharenborg, O., Boves, L., ten Bosch, L., in press. 'Early recognition' of polysyllabic words in continuous speech. Computer Speech and Language, in press, doi:10.1016/j.csl.2005.12.001.

Seneff, S., 2002. Response planning and generation in the Mercury flight reservation system. Computer Speech and Language 16, 283–312.

Seneff, S., 2004. The use of subword linguistic modeling for multiple tasks in speech recognition. Speech Communication 42 (3–4), 373–390.

Seneff, S., Wang, C., Hazen, T.J., 2003. Automatic induction of *N*-gram language models from a natural language grammar. In: Proceedings of Eurospeech, Geneva, Switzerland, pp. 641–644.

Sturm, J., Boves, L., 2005. Effective error recovery strategies for multimodal form-filling applications. Speech Communication 45, 289–303.

Tang, M., Seneff, S., Zue, V., 2003. Two-stage speech recognition using feature-based models: a preliminary study. In: Proceedings of the Workshop on Automatic Speech Recognition and Understanding, St. Thomas, US Virgin Islands, pp. 49–54.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L., 2000. Jupiter: a telephone-based conversational interface for weather information. IEEE Transactions on Speech and Audio Processing 8 (1), 85–96.