ACCURATE RECOGNITION OF CITY NAMES WITH SPELLING AS A FALL BACK STRATEGY

Josef G. Bauer Jochen Junkawitsch
Siemens AG, Corporate Technology
81730 Munich, Germany
{Josef.Bauer,Jochen.Junkawitsch}@mchp.siemens.de

ABSTRACT

Entering city names is an important issue for various speech driven applications such as telephone directory assistance. This paper proposes a system that combines word recognition with utterance verification and spelling as a fall back strategy. Word recognition experiments show that the use of a medium size vocabulary yields the lowest error rate when only permitting very few false acceptances. For letter recognition discriminatively trained letter models are applied. In order to shorten the spelling procedure two abort conditions are introduced which reduce the number of letters that have to be spelled. The system handles 96.5% of all calls correctly while less than 45% of all callers must spell the name.

1 INTRODUCTION

City name recognition over telephone lines is a challenging task with high interest for directory assistance services. As the number of different names can reach several ten thousands recognition performance of a word recognizer is generally insufficient for real world applications [1]. Although spelling is not as user-friendly as simply uttering the name, it is a feasible approach for this task because of its high accuracy [2]. This paper proposes an approach that combines word and spelling recognition in order to realize a user-friendly and accurate system. A similar method is considered in [3] but no evaluation is performed. This paper reports results of detailed experiments and gives advantageous extensions.

The paper is organized as follows: Section 2 describes the system architecture, section 3 gives an overview about the underlying feature extraction and HMM technology. In section 4 results of word recognition and utterance verification experiments are presented. Section 5 examines the use of spelling for city name recognition, whereas in section 6 both methods are combined. A final discussion is given in section 7.

2 SYSTEM OVERVIEW

The described system uses a word recognizer followed by an utterance verification procedure and an optional spelling recognizer as a fall back method for rejected utterances. In a first attempt the system tries to determine the correct name using a word recognizer. A simple confidence measure is used to verify the recognizer output. If the utterance is rejected the user is asked to spell the name letter by letter. In order to achieve a high user acceptance a very restrictive rejection for word recognition is chosen. A very low false acceptance rate is desired because the user might prefer to spell a name rather than being misunderstood by the system.

In the second stage of the system an isolated letter recognizer in combination with a spelling post processor allows highly accurate recognition of spelled city names. In addition two specific abort conditions are introduced that shorten the spelling procedure by interrupting the user. By this way, the average number of letters that have to be spelled can be decreased significantly.

3 BASIC RECOGNIZER TECHNOLOGY

Feature Extraction Telephone speech at 8 kHz sampling rate is filtered through a band pass and a preemphasis filter. Every 10 ms 24 mel scaled cepstral coefficients are computed by convolution of sinc kernels with the log fft spectrum of hamming windowed portions of the speech signal with a length of 25 ms. The cepstral coefficients as well as the window energies are filtered to compensate for linear channel distortions. First and second order derivatives of the energy parameter and of a reduced set of 12 cepstral coefficients are added to form a baseline feature vector with 51 components. A linear transformation based on Linear Discriminant Analysis is applied to a supervector formed by two subsequent feature vectors. The first 24 components of the transformed feature vector serve as input for HMM based classifiers. More details on the applied feature extraction can be found in [4].

HMM Technology The word recognizer and the spelling recognizer are based on Continuous Densities Hidden Markov Models. Phoneme and whole word models with Bakis topology use mixtures of Gaussian densities that model emission probabilities. Only one single global variance is estimated. Transition probabili-
ties are tied over all models and are set to fixed values allowing self loops, single transitions and one state skips. The mean vectors of the Gaussian densities are estimated with Viterbi Maximum Likelihood training and Minimum Classification Error training in case of whole letter models as described below. A system based on the same underlying HMM technology is described in [5].

4 WORD RECOGNITION

All word recognition experiments were carried out with general purpose phoneme HMMs. The SieTill database and the German SpeechDat(M)\textsuperscript{1} database (1700 speakers) that were recorded via public telephone networks were used to train triphone models with a total of 15k Gaussian densities.

A list of 22077 city names, that were obtained from the public telephone directory, was used to build a lexicon and a unigram language model for testing purposes. All unigram probabilities were chosen proportionally to the number of telephone ports in the city. A total number of 2329 utterances from 1610 different speakers were selected as a test set from the SpeechDat(II)\textsuperscript{1} database.

In a first experiment the dependence of recognition accuracy on the vocabulary size is examined. Therefore, all 22077 city names are sorted by their unigram probability and only the $n$ most probable entries are selected as the active vocabulary. Table 1 shows the OOV (Out-Of-Vocabulary) rate of the test set and the corresponding recognition accuracy for different vocabulary sizes. It can be seen that recognition accuracy saturates at about 85% even when a full vocabulary with no OOV words is applied.

In a second experiment utterance verification is performed in order to reject uncertain recognizer output. For this purpose a log likelihood ratio (LLR) is defined between the acoustic score of the recognized word and the best acoustic emission probability at every frame [6]. This LLR is accumulated within word boundaries and normalized by its length.

The rejection threshold for this confidence measure is determined by prescribing a false acceptance rate (i.e. acceptance of misrecognized city names) of 1%, 3%, and 5% respectively. In figure 1 the total rejection rate (correct rejection + false rejection) is plotted for different false acceptance rates. The curves form a clear minimum at a vocabulary size of 1500 words. This minimum of the total rejection rate also corresponds to a minimum of total error rate (total rejection + false acceptance), i.e. at a vocabulary size of 1500 most utterances can be recognized correctly. Moreover, the use of a vocabulary of only 1500 words instead of a full 22k words lexicon reduces computational costs significantly.

5 SPELLING RECOGNITION

Letter Models 2248 utterances with about 15000 letters from 1354 speakers in the German SpeechDat(M)\textsuperscript{1} and the SieTill database were selected to form a set of training patterns for 29 whole letter models. For some idioms like “scharfes s” (sharp s) that were rarely seen in the training material general phoneme models were applied. As a starting point Maximum Likelihood models with a total number of 3000 Gaussians for word models were trained. In addition discriminative training based on a Minimum Classification Error (MCE) objective function was performed ([5]). Although the training material consists of sequences of words (letters) classes used for MCE training are chosen to be words. For this reason the objective of this procedure is called Minimum Word Error (MWE). In order to apply this word based measure all utterances were segmented automatically in sections containing single words (letters) even if they were not spoken.

<table>
<thead>
<tr>
<th>vocabulary size</th>
<th>OOV rate</th>
<th>recognition accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>29.5 %</td>
<td>68.4 %</td>
</tr>
<tr>
<td>500</td>
<td>19.6 %</td>
<td>76.6 %</td>
</tr>
<tr>
<td>1000</td>
<td>11.6 %</td>
<td>82.5 %</td>
</tr>
<tr>
<td>1500</td>
<td>9.23 %</td>
<td>84.0 %</td>
</tr>
<tr>
<td>2000</td>
<td>7.77 %</td>
<td>84.5 %</td>
</tr>
<tr>
<td>5000</td>
<td>4.94 %</td>
<td>85.7 %</td>
</tr>
<tr>
<td>10000</td>
<td>3.13 %</td>
<td>85.5 %</td>
</tr>
<tr>
<td>22077</td>
<td>0.00 %</td>
<td>84.9 %</td>
</tr>
</tbody>
</table>

Table 1: Results of word recognition experiments without utterance verification: OOV rate and recognition accuracy for different vocabulary sizes.

\textsuperscript{1}For information about SpeechDat see the following URL’s: http://www.speechdat.org http://www.icp.grenet.fr/ELRA/home.html

\[ \text{Figure 1: Total rejection rate (correct rejection + false rejection) for a constant false acceptance rate of 1\%, 3\%, and 5\%.} \]
The performance of the letter recognizer was evaluated on speech sections containing only single letters (isolated mode as for MWE training) as well as on continuously spoken letters sequences as provided in the SpeechDat databases. For continuous recognition no language model was applied. Table 2 shows both isolated and continuous recognition results for an evaluation set consisting of 1563 utterances from 819 speakers of SpeechDat(II)\(^1\). These utterances consist of spelled proper names and random letter sequences.

Table 2: Word error rates (WER) in % for isolated and continuous letter recognition (IWR, CWR) with phoneme based models (PHONE), word based models (WORD) and additional discriminative training (MWE). For continuous recognition word deletion and insertion rates in % are given: DELS, INS.

<table>
<thead>
<tr>
<th>HMMs</th>
<th>IWR WER</th>
<th>CWR WER</th>
<th>DELS</th>
<th>INS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHONE</td>
<td>19.7</td>
<td>31.7</td>
<td>2.3</td>
<td>5.9</td>
</tr>
<tr>
<td>WORD</td>
<td>16.3</td>
<td>21.5</td>
<td>1.7</td>
<td>3.1</td>
</tr>
<tr>
<td>WORD+MWE</td>
<td>12.5</td>
<td>16.0</td>
<td>1.9</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Spelling with Post Processing According to the experiment described above the output of a continuous letter recognizer features a considerable numbers of insertions and deletions. For this reasons isolated letter recognition with prompting for each letter is applied in the described spelling system. By this means the spelling post processing does not have to deal with a word graph. Instead the post processing module receives a n-best list with letter hypotheses and HMM probabilities for each spoken letter. With respect to the list of all possible city names (letter chains) it computes the probability that a specific city name was spelled. The regular lattices produced by isolated letter recognition also allow an easy computation of probabilities for preliminary results while spelling of a name is still in progress. A tree organization of the search space and a beam search allow processing for several ten thousand names in a small fraction of real time.

Having no speech material with spelled city names available the performance of the spelling system was evaluated with the help of a simulation strategy. Speech sections containing only single letters (as used for discriminative training) were assembled randomly to form chains of letters corresponding to names from the list of all city names. Isolated letter recognition is then performed within the boundaries of each speech section. It should be noted that this strategy does not consider letter insertions or deletions that can occur in real world systems even when letters are spoken in an isolated manner.

Speech segments were taken from the same material that was used for the results shown in table 2. Table 3 shows the performance of the spelling module. In these experiments the simulation strategy was applied once to each of the 22077 city names. For this and all further spelling experiments the name probabilities from the unigram language model were not taken into account.

In the described experiments pruning was always adjusted in such a way that no extra errors were introduced because of the sub-optimality of the beam search. Even with this kind of conservative pruning in many cases there remains only one possible name hypothesis (preliminary result) after processing several letters. From this observation a first abort condition (abort condition 1) is derived that possibly allows to abort the spelling process after several letters. Analysis of abort condition 1 shows that the mean number of letters that have to be spelled can be reduced from 11.0 (mean name length in letters) to 8.13 in the simulation described above.

### Table 3: Performance of the spelling system with different letter models (see table 2).

<table>
<thead>
<tr>
<th>HMMs</th>
<th>rank 0</th>
<th>rank 0 - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHONE</td>
<td>96.3</td>
<td>98.0</td>
</tr>
<tr>
<td>WORD</td>
<td>97.1</td>
<td>98.8</td>
</tr>
<tr>
<td>WORD+MWE</td>
<td>97.8</td>
<td>98.8</td>
</tr>
</tbody>
</table>

6 COMBINATION OF SPELLING AND WORD RECOGNITION

For the following experiments a vocabulary size of 1500 words together with a rejection strategy resulting in a 3% false acceptance rate for word recognition is chosen. This gives an absolute number of 1041 utterances from the word recognition test set (see section 4) that are rejected in the first stage of the system.

Performing the spelling simulation for this list of names an accuracy of 98.9% was achieved with the most advanced acoustic models (WORD+MWE). For this 1041 names an average number of 8.0 letters have to be spelled without an abort condition. Applying the abort condition 1 the mean number of letters that have to be spelled could be reduced to 6.6.

Looking at the spelling post processing it can be found that the beam search results in a finite list \(V\) of preliminary spelling results (city names) that can be derived after each spelled letter. Now a second abort condition is introduced that is based on the combination of word recognition and spelling. It is assumed that after each spelled letter the HMM probability for the spoken city name is
known for all words in $V$. This can either be achieved by a n-best search for the full vocabulary in stage 1 or by an n-best search with vocabulary $V'$ that is performed after several letters are spelled. Now for the abort condition 2 the probability ratio of the two most probable word hypotheses in $V$ is compared with a rejection threshold. If this ratio exceeds the threshold the spelling process can be aborted (*abort condition 2*).

Again for the following experiments this rejection threshold was set to a very conservative value causing no extra errors due to the abort condition. Figure 2 shows how the two abort conditions are able to reduce the number of letters that have to be spelled. For the 1041 city names the mean number of letters that have to be spelled can be reduced to 4.9. From figure 2 it can be seen that in most cases full spelling of long words can be avoided with the help of the abort conditions.

![Figure 2: Percentages of accepted names with and without abort conditions after spelling of a certain number of letters.](image)

### 7 DISCUSSION

The advantage of the proposed system is a good compromise of an user-friendly behavior and a high recognition accuracy considering the high perplexity of this city name recognition task.

In a first stage word recognition and utterance verification is performed with a strong rejection threshold in order to accept only recognizer output with a high degree of confidence. The use of a medium size vocabulary of 1500 words yields two advantageous effects for word recognition. First, the percentage of correctly accepted city names has a maximum at this medium vocabulary size. Second, the computational requirements are essentially lower than for a full vocabulary size of 22077 words.

In a second stage spelling recognition is performed for the rejected utterances. In order to improve the user-friendly behavior two specific abort conditions are introduced that interrupt spelling as prematurely as possible. By this way the average number of spelled letters could be reduced from 8.0% to 4.9%.

Table 4 shows the overall performance of the complete system. The total number of calls that could be handled correctly in our simulation was 96.5%. For this setup 55.3% of all calls could be handled without spelling. A weaker rejection threshold for word recognition leads to smaller numbers of callers that have to spell. In this case the higher number of errors in word recognition has to be compensated by an intelligent dialog with system queries for uncertain cases.

As user acceptance must be the final quality criterion for a dialog system only field trials would allow to optimally adjust and expand the proposed system in terms of rejection parameters and dialog design.

### REFERENCES


<table>
<thead>
<tr>
<th></th>
<th>Success in word recognition (stage 1)</th>
<th>Errors in word recognition (stage 1)</th>
<th>Success in spelling (stage 2)</th>
<th>Errors in spelling (stage 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>52.3%</td>
<td>3.0%</td>
<td>44.2%</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

Table 4: Successfully handled calls and errors occurring in word recognition (stage 1) and spelling (stage 2).