TOWARDS AN AUTOMATED DIRECTORY INFORMATION SYSTEM

Frank Seide[+] *and Andreas Kellner*[*]

[*]Philips GmbH Research Laboratories Aachen, P.O. Box 500145 D-52085 Aachen, Germany
[+]Philips Research Laboratories Taipei, P.O. Box 22978, Taipei, Taiwan, R.O.C.
E-mail: {seide@prlt,kellner@pfa}.research.philips.com

ABSTRACT

This paper describes a design and feasibility study for a large-scale automatic directory information system with a scalable architecture. The current demonstrator, called PADIS-XL[1], operates in realtime and handles a database of a medium-size German city with 130,000 listings.

The system uses a new technique of taking a combined decision on the joint probability over multiple dialogue turns, and a dialogue strategy that strives to restrict the search space more and more with every dialogue turn.

During the course of the dialogue, the last name of the desired subscriber must be spelled out. The spelling recognizer permits continuous spelling and uses a context-free grammar to parse common spelling expressions.

This paper describes the system architecture, our maximum a-posteriori (MAP) decision rule, the spelling grammar, and the dialogue strategy. We give results on the SPEECHDAT and SIETILL databases on recognition of first names by spelling and on jointly deciding on the spelled and the spoken name. In a 35,000-names setup, the joint decision reduced name-recognition errors by 31%.

1. INTRODUCTION

The use of speech recognition for an automated telephone directory information service offers a large potential for automation and increased functionality. In this paper, we present PADIS-XL, a design and feasibility study of such a system that handles the directory of the city of Aachen, Germany, and vicinity. Its 131,001 listings include 38,608 distinct last names, 9938 first names, and 2049 streets. Our main focus was not primarily on usability issues, but on how to handle the very large search space with today's speech recognition technology.

PADIS-XL is based on our PADIS system [1], an automatic switchboard system that allows a free, mixed-initiative dialogue in spontaneously spoken German, aiming at environments of some thousand subscribers. However, when scaling it up to name lexica covering cities or even countries, the computational effort becomes prohibitive, since the recognizer would have to be able to recognize all possible names at any time.

Thus, we chose a system-driven dialogue for PADIS-XL, in which the caller may answer only exactly one word (spelled or spoken) per dialogue turn. The recognizer's lexicon is switched, enabling only those words expected at a turn.

[1]PADIS: Philips Automatic Directory Information System.

The dialogue aims at reducing the search space with every dialogue turn. This strategy is in line with the one used in the 5000-entry FAUST demonstrator presented by *Kaspar et al.* [2] and the Ipswitch field-test system described by *Whittaker and Attwater* [3].

In addition, for such a large system, acceptable recognition accuracy can only be obtained by consequently using all available knowledge sources. During the dialogue, the caller is asked to spell the last name, to speak it, and to say the first name and the street. These items are highly redundant, which we effectively exploit in our approach.

In [4], we incorporated database constraints and dialogue history into the MAP decision criterion of our PADIS system to improve the within-turn error rate. For PADIS-XL, we followed the same idea, but since we now collect only one item per turn, we apply the database constraint to the *whole dialogue* and take the final decision about the caller's dialogue goal jointly on *all dialogue turns*.

The paper is organized as follows. Section 2 explains the system architecture. In section 3, we describe the MAP decision criterion. Section 4 covers the spelling module, and section 5 the dialogue strategy. In section 6, we present offline results on recognition of spelled and spoken first names. Our conclusions are given in section 7.

2. SYSTEM ARCHITECTURE

Figure 1 shows the system architecture. The system consists of a speech recognizer, a special spelling postprocessor, a speech-understanding and dialogue-control component, and a speech-output unit.

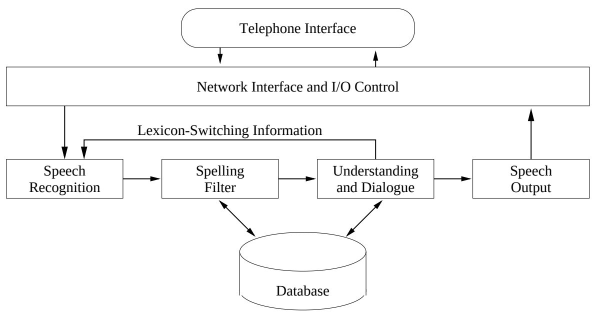

Figure 1: System architecture.

As the interface between speech recognition, spelling, and speech understanding, we use word graphs [5]. A word graph is a compact representation of plausible alternative sentence hypotheses. Every path through the graph is a sentence hypothesis.

At every dialogue turn, the *speech recognizer* processes the caller's utterance and produces a word graph. In spelling mode, this is a letter graph. Otherwise, the recognizer is configured to allow single-word utterances only, so a word graph then merely represents a single-word candidate list. To achieve realtime operation, our recognizer's vocabulary can now be switched enabling only those words expected in the current dialogue state.

The *spelling-filter* module scans the recognizer's output for letter sequences that form valid names according to a name list. Those names are then added as word hypotheses to the word graph. Thus, the subsequent speech-understanding engine smoothly integrates with spelling.

In PADIS-XL, *speech understanding and dialogue control* are much more closely integrated than in our previous systems [1, 6]. Speech understanding became trivial due to the single-word restriction. However, dialogue control had to be extended to keep multiple hypotheses and to take the final joint decision.

An issue for PADIS-XL is *speech output*. Our former approach of replaying prerecorded phrases is obviously not feasible. Instead, a text-to-speech system must be used.

3. DECISION RULE

The basic idea is to apply the decision rule we derived in [4] jointly to all dialogue turns. In PADIS-XL, M information items are collected in M turns, with $M \leq 4$, as shown in table 1.

Table 1: The four information items.

I_1	the last name spelled
I_2	the last name spoken
I_3	the first name (spoken)
I_4	the street (spoken)

Simplified by the system-driven single-word approach, the derivation of the decision rule is straightforward:

We define the speech understanding task as finding the directory listing defined by the information item set $\hat{I} = \{\hat{I}_1, \hat{I}_2, ..., \hat{I}_M\}$ that was most probably the one the user uttered when he generated the acoustic observations $O = O_1, O_2, ..., O_M$ in the M turns (MAP criterion):

$$\begin{aligned} \hat{I} &= \arg\max_I P(I|O) \qquad (1) \\ &= \arg\max_I p(OI) \end{aligned}$$

We introduce $W = W_1, W_2, ..., W_M$ as the underlying word sequences that are used to utter the information items $I_1, I_2, ..., I_M$. W_1 is a spelled-letter sequence, while all other W_i are single words.

$$\begin{aligned} \hat{I} &= \arg\max_I \sum_W p(OIW) \\ &= \arg\max_I \sum_W p(O|WI) \cdot P(W|I) \cdot P(I) \end{aligned}$$

To compute $P(I)$, we introduce the dialogue goal (directory listing) $G = G_1, G_2, ..., G_M$, with G_i corresponding to I_i. Since G is unknown, we sum over all possible values:

$$\hat{I} = \arg\max_I \sum_W p(O|W) \cdot P(W|I) \cdot \sum_G P(I|G) \cdot P(G)$$

The prior $P(G)$ reflects how likely the listing G is asked for. This information should be provided by the underlying database; in PADIS-XL, we assume it equal for all listings. The switch $P(I|G)$ is 1.0 if all I_i match their respective G_i, and 0 otherwise. Approximating the sum over W by the maximum, we obtain the final decision rule:

$$\begin{aligned} \hat{I} \approx \arg\max_I \Bigg\{ & \max_W \underbrace{\prod_{i=1}^{M} p(O_i|W_i)}_{\substack{\text{acoustics}\\ \text{(recognizer)}}} \cdot \underbrace{\prod_{i=1}^{M} P(W_i|I_i)}_{\substack{\text{grammar}\\ \text{(spelling)}}} \\ & \cdot \underbrace{\sum_G P(I|G) \cdot P(G)}_{\substack{\text{database knowledge}\\ \text{(dialogue control)}}} \Bigg\} \qquad (2) \end{aligned}$$

For the spelled item, $P(W_i|I_i)$ is delivered by the spelling grammar, while for the spoken single-word items, it is simply 1.0 if W_i is the pronunciation of I_i and 0 otherwise.

4. SPELLING FILTER

The spelling filter is a new component in the Philips automatic inquiry system. It acts as a postprocessor to the speech recognizer. For every utterance, it reads a word graph from the recognizer containing spelled letters and other words that are used in spelling expressions (e.g. "*double*"). Figure 2 shows such a word graph for the user input "*M. I. double L. E. R.*". As its output, the spelling module creates an extended word graph that contains all spelled words as word hypotheses.

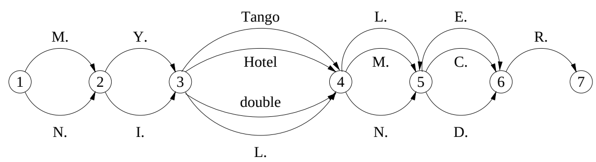

Figure 2: Sample word graph for *"M. I. double L. E. R."*.

In spontaneous spelling, the users do not always spell a name letter by letter. Instead, they also use descriptive phrases like "*double T.*" or "*M. as in Mike*". In order to handle such expressions, the spelling module operates in a two-stage process:

In the first stage, descriptive phrases in the input are identified and translated into letters or letter strings. This is done by parsing the word graph with an attributed stochastic context-free grammar like the one used in our speech-understanding systems [6]. The grammar contains rules for typical descriptive phrases like common spelling alphabets and expressions like *"double T"*, *"A. Umlaut"*, or *"Y. as in Yankee"*, and also for the plain letters.

In a scenario where the caller is not restricted to either spell or speak a name in separate turns, the grammar would also handle expressions like "*Meyer with Y.*".

The result of the parse is stored in a search graph (figure 3). It has the same nodes as the underlying word graph, its arcs are the letters or letter sequences created from the letters and descriptive expressions.

In the second stage, an additional knowledge source is incorporated: a word list containing all valid last names, 38,608 in our demonstrator. The graph is searched for letter sequences that form valid names, and for every valid spelled word found, a word arc is added to the letter graph. In our example graph, the names *Miller*, *Mitler*, and *Milner* would probably be considered valid (depending on the name list).

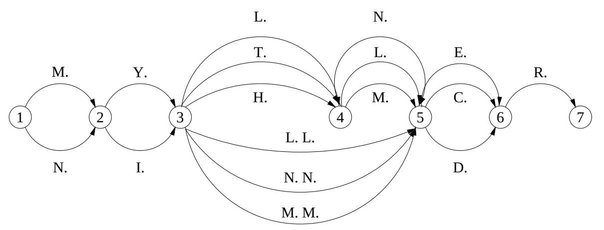

Figure 3: Sample search graph.

The new word-arc's score is computed from the acoustic likelihoods of the underlying letter sequence, $p(O_i|W_i)$, and the language-model probability $P(W_i|I_i)$ delivered by the stochastic spelling grammar [6].

5. DIALOGUE STRATEGY

The purpose of the dialogue is to gather the information items required to find the desired database entry. The user is asked for the four information items shown in table 1 in the order shown. Since our current system is limited to Aachen and vicinity, the city name is not asked for.

The dialogue is system-driven and strives to reduce the search space (the set of possible listing candidates) with every dialogue turn. It terminates as soon as the search space has been reduced to three directory listings or less.

In the first turn, the user is asked to spell out the desired last name. At that time, the search space consists of the full database, but the recognizer is limited to spelling. The spelling filter identifies proper spelled last names in the recognizer's output.

Due to the acoustic pruning (beam search) in letter-graph generation, only a small subset of the possible last names is contained in the letter graph. Furthermore, candidates classified as unreliable according to a confidence measure are also discarded. The number of surviving last names is usually significantly less than 100.

In the subsequent dialogue turns, the user is asked to speak the last name, the first name, and finally the street name, one after the other. The recognizer is dynamically configured to recognize only those words (last names, first names, or streets, respectively) that refer to one or more of the possible candidates identified in the previous turns. Then, for every recognized word hypothesis, the path score is combined with the scores of the corresponding candidates, forming a new candidate list with a joint probability assigned to each candidate. Again, candidates not found anymore in the recognizer's output due to pruning, as well as unreliably recognized candidates, are deleted from the search space.

As soon as the list contains three or less candidates, the strategy ends, and the user is asked to confirm one by one sorted by score. Finally the desired phone number is presented.

The approach automatically handles symbolic and acoustic ambiguity. For example, a last name is *symbolically* ambiguous if more than one person with that name exists. It is *acoustically* ambiguous if acoustically similar names exist that cannot be disambiguated correctly by the recognizer without additional information.

6. EXPERIMENTAL RESULTS

We have not conducted a field test yet, so no real-life test data was available, and dialogue success rates cannot be given. Instead, we evaluated our recognizer as well as the performance gain from our joint-decision rule on first-name recognition. This task is largely comparable to our online system and could be tested on the two German telephone databases SPEECHDAT [7] and SIETILL.

6.1. Corpus Description

The spelling subcorpus of SPEECHDAT contains spelled first names, words, and random letter sequences. In SIETILL, speakers were requested to both spell and speak their first names. Only plainly-spelled utterances were used, expressions and spelling alphabets were excluded.

For letter recognition, whole-word models were trained on 1637 SPEECHDAT sentences (1.2h non-silence). For spoken names, we used the triphone models from PADIS, trained on 33,081 utterances (12.1h non-silence) of spontaneous train-schedule inquiries [4, 6]. The letter-bigram model used in letter-graph generation was trained on the first names of the 1995 phone book of Aachen.

Table 2: Evaluation corpora.

	SPEECH-DAT	SIETILL spelled	SIETILL spoken
word units	2899	3557	581
utterances	357	581	581
length [h:min]	0:48	1:36	0:19
letter-bigram PP	15.7	11.0	

For evaluation, we used the remaining spelled names and words of SPEECHDAT and those SIETILL sentences for which a spelled as well as a spoken version was available, see table 2.

6.2. Spelling Results

We investigated how the error rate depends on the number of names to be distinguished. For every utterance, a letter graph was created using the word-graph algorithm described in [8] (word-conditioned tree copies). Then, the best path through the graph was extracted, considering only paths consisting of exactly one single proper name as defined by a name list.

Table 3 shows spelled-word error rates (WER) and letter error rates (LER) for name lists of different size. The first line shows the spelled-word and letter graph-error rates (GER) [5], i.e. the lower bound imposed by pruning errors during lattice generation. The average graph density was about 9 hypotheses per spoken letter.

First-name lists were extracted from the 1995 phone books of two different German cities, Aachen (pop. 250,000; 8895 different first names) and Hamburg (pop. 1.7 million; 35,335 first names). The "Hamburg/n" lists contains only every n-th entry of the original list. Since out-of-vocabulary problems were out of the scope of this investigation, we added all missing words from test corpora to

Table 3: Spelling-error rates over name-list size.

name list	list size	SPEECHDAT WER	SPEECHDAT LER	SIETILL WER	SIETILL LER
GER	–	6.2%	1.1%	7.8%	1.6%
test-set words	414	7.3%	7.1%	9.3%	8.4%
Aachen	8895	10.9%	6.9%	17.4%	8.2%
Hamburg/4	9151	9.5%	7.0%	13.6%	7.5%
Hamburg/3	12052	10.4%	6.9%	15.2%	8.3%
Hamburg/2	17887	11.5%	7.2%	16.2%	7.4%
Hamburg	35335	13.7%	7.7%	20.8%	8.4%
letter bigram	∞	68.9%	17.6%	62.1%	19.4%

the name lists. In a control experiment ("test-set words"), the vocabulary consisted of exactly these words.

The line termed "letter-bigram" shows the error rates using the letter-bigram model instead of the name-list constraint.

6.3. Combining Spelled and Spoken Utterances

To assess the gain obtained by the joint-decision rule as given in eq. (2), we used the SIETILL database. Here, 581 speakers uttered a plainly spelled and a spoken version of her/his first name. This permitted us to assess the effect of joint decision when applied to $M = 2$ information items. We chose the 35K Hamburg first-names list, because it is comparable in size to the last-name list of Aachen (38K) used in our demonstrator.

Including pronunciation variants, the base vocabulary size was 37,961. To speed up search, we dynamically switched the recognizer's lexicon for each utterance to use only the subset of words found in the corresponding letter graph. This approach, which is also used in the demonstrators described in [2, 3], led to an effective lexicon size of on average only 15.3 (max. about 200), permitting realtime operation.

Table 4 shows the results. On the 35K name list, the spelling recognizer alone achieves a spelled-word error rate of 20.8%.

The lexicon-switched spoken-name recognizer obtained a significantly worse error rate of 27.7%. However, this error rate seems to depend extremely on the actual lexicon size and is very unstable. In an earlier experiment with an average lexicon size of 5.6, the spoken recognizer performed about rel. 5% *better* than the spelling recognizer.

Table 4: Results on joint decision.

recognition mode	all utterances WER	all utterances gain	excl. graph errs WER	excl. graph errs gain
spelled-word GER	7.8%	–	(0.0%)	–
spelling	20.8%	0%	(14.2%)	0%
spoken	27.7%	-33%	(21.6%)	-52%
joint decision	14.3%	31%	(7.1%)	50%

When we used the joint-decision rule, taking both the spelling and the spoken probabilities into account, a significant gain of 31% was achieved.

7.8% of the errors are caused by search errors in letter-graph generation (45 utterances). To separate the joint-decision effect from these search errors, we recomputed the error rates excluding these 45 utterances. This is also shown in table 4.

7. CONCLUSIONS AND FUTURE WORK

We have described our design and feasibility study PADIS-XL, a large-scale automatic directory information system that handles a 130K directory in realtime. Its new features are a system-driven dialogue that strives to reduce search space from turn to turn, dynamic lexicon switching to keep the recognition effort limited, the use of spelling, and a joint-probability decision criterion that combines all user's utterances to achieve acceptable recognition accuracy.

We have also described offline results on name recognition on the German SPEECHDAT and SIETILL databases. We found spelling in combination with dynamic lexicon switching an efficient way to obtain realtime spoken-name recognition. However, the spoken-name recognizer performed significantly worse than the spelled-name recognizer. On the other hand, our new decision criterion achieved a relative improvement of 31% over spelled-name recognition using the joint probability of the spelled and the spoken utterance. This resulted in a name error rate of 14.3% for a 35K name list. We believe the gain for the online demonstrator to be even larger because it takes up to four turns into account.

Future work will include a field test to study usability issues and their consequences for the system design. The field test will also permit us to verify the results on a much larger corpus.

After all, we believe that automating simple directory-assistance requests, in which the caller knows all required information, will become feasible in the near future.

8. REFERENCES

1. A. Kellner, B. Rueber, and F. Seide. A voice-controlled automatic telephone switchboard and directory information system. In *Proc. IVTTA*, pp. 117–120, Basking Ridge, Sep. 1996.
2. B. Kaspar, G. Fries, K. Schuhmacher, and A. Wirth. FAUST – A directory-assistance demonstrator. In *Proc. Eurospeech*, pp. 1161–1164, Madrid, Sep. 1995.
3. S.J. Whittaker and D.J. Attwater. Advanced speech applications – the integration of speech technology into complex services. In *ESCA workshop on Spoken Dialogue Systems – Theory and Application*, pp. 113–116, Vigsø, Jun. 1995.
4. F. Seide, B. Rueber, and A. Kellner. Improving speech understanding by incorporating database constraints and dialogue history. In *Proc. ICSLP*, Vol. 2, pp. 1017–1020, Philadelphia, Oct. 1996.
5. M. Oerder and H. Ney. Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *Proc. ICASSP*, Vol. II, pp. 119–122, Minneapolis, Apr. 1993.
6. M. Oerder and H. Aust. A realtime prototype of an automatic inquiry system. In *Proc. ICSLP*, Vol. 2, pp. 703–706, Yokohama, Sep. 1994.
7. http://www.phonetik.uni-muenchen.de/SpeechDat
8. H. Ney and X. Aubert. A word-graph algorithm for large-vocabulary continuous-speech recognition. In *Proc. ICSLP*, Vol. 3, pp. 1355–1358, Yokohama, Sep. 1994.