Strategies for Name Recognition in Automatic Directory Assistance Systems

Andreas Kellner, Bernd Rueber, and Hauke Schramm

Philips Research Laboratories
Weisshausstrasse 2, 52066 Aachen, Germany
{kellner, rueber, schramm}@pfa.research.philips.com

ABSTRACT

Recognition of large numbers of different names is the central problem in automatic directory assistance services and many other applications for spoken language dialogue systems. This paper investigates a methodology of stochastically combining N-best lists retrieved from multiple user utterances with the telephone database as an additional knowledge source.

This strategy is used in a prototype of a fully automated directory information system which is designed to cover a whole country: After the city has been selected, the user is asked to spell and say the name of the desired person and if necessary also the first name and street. The number of active database entries is reduced in every turn until only a single database entry is left.

Results for different recognition strategies are presented on a real-life data collection for databases of various sizes with up to 1 million entries (city of Berlin). The experiments show that a substantial part of all simple requests can be automated with the strategy presented (>80% correctly recognized, 10% rejected).

I. INTRODUCTION

In recent years, the task of automating directory assistance has generated great interest in the scientific community. Thus, several demonstrator systems have been set up [1, 2, 3] and some field trials were performed [1, 4]. Besides, several groups presented quantitative results on recognition performance for various directory sizes and knowledge sources [4, 5, 3, 6, 7].

Nevertheless, there still is no solution for the complete automation of directory assistance requests for a whole country, and there are no thorough systematic results on what is the relative value of using all available knowledge sources. E.g. some studies restricted themselves to combinations of spelled and spoken last names while others only performed investigations for a single database size.

This paper starts out from the demonstrator system for a fully automated directory information for the city of Aachen with 131.000 database listings [3]. Based on this work, we created a prototype system which, by its hierarchical structure, can handle a complete country.

A dialogue example from this system is shown in figure 2. In the course of the dialogue, the system takes a combined decision on the joint probability over multiple dialogue turns, using the directory database itself as additional knowledge source [3]. In this way, the search space, which consists of all 'active' database entries can be reduced step by step.

In section II of this paper we present an overview of the system and its components and describe the dialogue design. In Section III, we present a systematic evaluation of this approach, i.e. a comparison of error rates for using joint (redundant) information with and without dynamic lexicon switching.

II. SYSTEM OVERVIEW

A. System Architecture

The prototype system consists of a speech recognizer, a spelling filter, a dialogue manager, and a text-to-speech module as shown in figure 1. A language resource manager which is controlled by the dialogue manager provides the speech recognizer and the spelling filter with the active vocabulary for the current dialogue situation.

Word graphs are used as interface between speech recognition, spelling filter and dialogue control. The speech-understanding module which interprets the input and detects user commands is integrated in the dialogue-manager module.

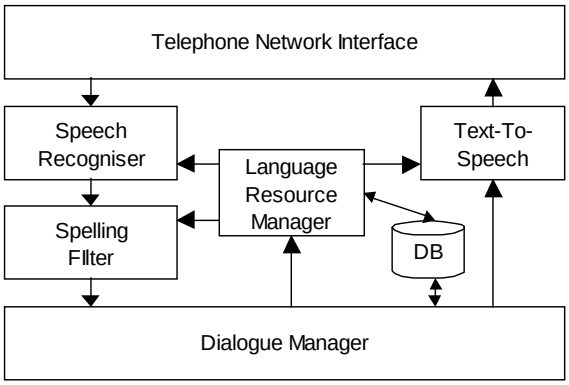

Fig. 1- System Architecture

B. Speech Recognizer

The recognition experiments presented in this paper were obtained, using a speaker independent telephone-speech decoder. This state-of-the-art continuous density HMM recognizer works with two different setups for the recognition of spoken respectively spelled words. The switching between these setups is done by the language resource manager which also delivers a list of active words for the recognition of single words.

B.1. Single Word Recognition

In this mode, the decoder is restricted to the recognition of a single word per utterance. An inventory of 3502 strongly tied context-dependent phonemes was used. Due to a lack of appropriate (i.e. isolated word) speech data, we trained this phoneme set on a large German spontaneous speech database, consisting of 33081 utterances (12.1h non-silence) of train-schedule inquiries. This mismatch causes, of course, an increase in word error rate. But this is judged to be negligible for the basic results of our experiments.

B.2. Spelling Recognizer

In order to make the recognition of spelling words like '*double'* possible, our spelling recognizer worked with a phoneme set containing two subsets. The first one consisted of the phonemes used for the isolated word recognition and the second comprised 61 context dependent spelling phonemes. The latter were trained on 1637 spelled first names, words and random letter sequences (1.2h non-silence frames), taken from the German telephone database SPEECHDAT [8].

C. Spelling Filter

To handle very large lists of proper names, spelling is essential. Furthermore our studies have shown, that in real-life situations, people do not always spell a name letter by letter. Instead, they also use descriptive phrases like '*double T'* or '*M as in Mike'*.

In our system, we use a spelling filter which acts as a postprocessor to the speech recognizer [3]. The recognizer delivers a word graph containing spelled letters and other words which are used in spelling expressions.

In a first stage, the word graph is parsed for descriptive phrases using a context-free grammar which describes the most common phrases and spelling alphabets. The result of this parsing is stored in a search graph in which the descriptive phrases are replaced by the generic letter sequence they describe.

In the second stage, a background wordlist generated from the database is used to identify valid names in the search graph. These words are stored in a new word graph which is sent to the dialogue module. The score of each word in this graph is composed from the underlying acoustic scores and a language model score delivered by the spelling grammar.

D. Dialogue Strategy

The prototype system follows a hierarchical dialogue strategy (cf. Fig.2): In the first step, the system asks for the city. At this point, only a limited vocabulary containing the largest cities is activated in the recognizer. If none of those cities was understood with sufficient reliability (using the reliability measure from [9]), the user is asked to spell the city name. When the user has verified the city, the database of the selected city can be activated.[1]

Now, the dialogue aims at reducing the number of active database entries with every turn. In the beginning, the search space consists of all directory listings of the selected city. The user is asked to spell out the desired last name. The search space is then reduced to only those database entries for which the name was found in the spelling graph. In the subsequent dialogue turns, the recognizer is dynamically configured to recognize only those words (last names, first names, or streets, respectively) that refer to active database entries.

System:	Hi, this is the automated directory information. From which city do you want to have a listing?
User:	*Aachen.*
System:	Do you mean Aachen?
User:	*Yes.*
System:	Please spell the last name of the desired person.
User:	*M I double L E R.*
System:	Please say the last name.
User:	*Mille.r*
System:	Please say the first name.
User:	*John.*
System:	Do you mean John Miller, Cedar-Street?
User:	*Yes.*
System:	The telephone number is .... Should I put you through? ...

Fig. 2 – Dialogue Example

In each turn, the scores of the recognized hypotheses are combined with the scores obtained so far for the corresponding database candidates. This forms a candidate list with a joint probability assigned to each candidate.

Due to pruning unreliably recognized candidates are deleted. Thus, the search space is reduced in every turn.

In some situations the correct hypothesis may not be in the n-best list delivered by the recognizer (e.g. due to pruning errors). In such cases, the n-best list is often disjoint with the active database subset, thus the combination of the two leads to an empty set of active candidates. This can be used as a safe rejection criterion. In such a case, the call may be forwarded to a human operator.

As soon as there are only three or less candidates left in the search space, these candidates are presented to the user.

At every point in the dialogue, the user can ask for (context-specific) help, for a restart of the dialogue, or for a human operator. While the dialogue is in principle system directed, i.e. the user has to answer the question stated by the system, this allows a minimum of user initiative.

[1] In our current prototype, database switching is not yet implemented. The system can therefore only be used for a single city.

The dialogue flow can be configured with a simple C-like dialogue description language which is based on Philips' HDDL [10].

III. RECOGNITION RESULTS

In this section, the recognizer's ability for the task of large scale directory assistance will be systematically assessed. For that, after explaining the general setup, in a first subsection, we present plain word error rates versus lexicon size, thus showing the clear necessity for joint recognition. Then, the next subsection shows the results of the combined recognition experiments.

A. Recognition Experiment Setup

A telephone database of directory assistance inquiries comprising 676 different speakers all over Germany has been collected in the following manner: By various advertisements people were asked to call up a data collection system which prompted them for speaking and spelling their last, first, street, and city names.

This data was used as test set in our experiments. Artificial telephone directories of varying sizes were created using the telephone directory of Berlin, Germany's biggest city with about 1.3 million database entries, in the following way:

1. All directories include the test data.
2. Then, different percentages of Berlin were added to them as a background list by selecting every n-th entry of the original Berlin directory. So, e.g. "0.1% of Berlin" consists of the test data plus every 1000-th entry of Berlin.

B. Word Error Rates versus Lexicon Size

Tables 1 – 3 and Figures 3 – 4 give the word and graph error rates versus the lexicon size for several parts of Berlin for last, first, and street name category.

There are some remarkable points about these numbers:

1. The word error rates, even for medium size lexica, are much to high to appear useful for a practical application.
2. The word error rates on street names are significantly better than those on last or first names. Thus, street names appear to be acoustically more discernible.
3. The graph error rates, i.e. the percentages of word graphs not containing the spoken item, of course, depend on the recognition details such as pruning thresholds. But even with fixing pruning thresholds, the graph densities, i.e. the average number of words in the word graph per spoken word, vary a lot. As a consequence, the graph error rates are extremely noisy. Nevertheless, for bigger tasks, the problem of generating word graphs of high enough quality is substantial. Therefore, in order to successfully apply a multiplicity of knowledge sources, already the first recognition step needs careful consideration.

Summarizing, it may be agreed

1. that the task of name recognition calls for the combination of more than one knowledge source,
2. that a useful recognition setup should start with the subset with the smallest and best discernible vocabulary, which are the street names in this case (at least as long as spellings are not considered).

% Berlin	lexicon. size	WER	GER
0.1	1824	39.2	4.3
1	10099	56.8	11.8
10	56993	73.1	38.0

Tab. 1 – Word Error Rate WER and Graph Error Rate GER versus lexicon size for last name recognition

% Berlin	lexicon. size	WER	GER
0.1	798	30.8	4.0
1	2502	41.57	4.1
10	11237	58.0	8.4

Tab. 2 – Word Error Rate WER and Graph Error Rate GER versus lexicon size for first name recognition

% Berlin	lexicon. size	WER	GER
0.1	1691	25.6	8.3
1	4714	35.4	11.5
10	7719	37.9	11.7

Tab. 3 – Word Error Rate WER and Graph Error Rate GER versus lexicon size for street name recognition

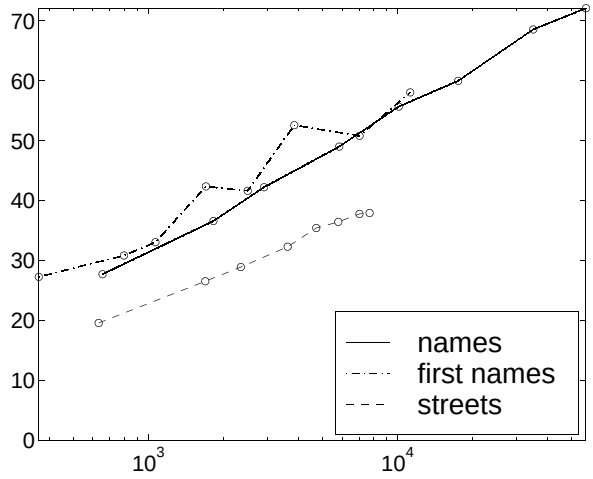

Fig. 3 – Word Error Rates versus lexicon size for last, first, and street name recognition

C. Joint Recognition

To improve the name recognition performance the following alternative joint recognition scenarios were studied [cf. 7 and 3]:

1. *SEP:* separately recognizing each name category for generation of n-best lists which are only afterwards combined,
2. *SEP*:* same as 1, but (as a control experiment to assess the importance of pruning errors) always artificially adding the spoken word to the word graphs (by using forced alignment),
3. *HIER:* hierarchical recognition, i.e. starting out with the recognition result of one name category, successively restricting the active lexicon for all

subsequent recognition steps as to include only the candidates left over so far.

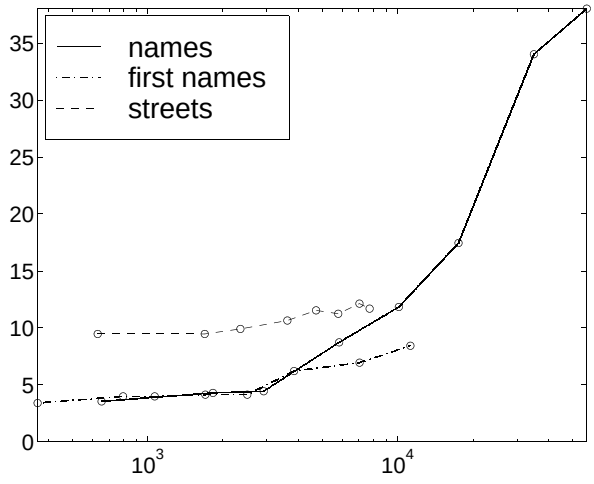

Fig. 4 – Graph Error Rates versus lexicon size for last, first, and street name recognition

In all these scenarios, combined n-best lists were computed by a standard weighted score addition: Let sc_1 be the score of an item in n-best list 1 and sc_2 the score of its matching entry in n-best list 2, i.e. the one where the combination of the two refers to a valid database entry. Then the score $sc_{1,2}$ of the combined entry in the combined n-best list is computed by

$$sc_{1,2} \quad sc_1 \quad \alpha \cdot sc_2 \qquad (1)$$

The weighting factor α has been optimized on a cross-validation corpus and α=1 turned out to be a reasonable choice.

In cases where several entries of the two n-best lists to be combined match each other, e.g. if a person is listed with two different first names, we select the best scoring entry. I.e., for each entry of list 1, the matching entry of list 2 is selected which has the lowest score.

For the recognition setup, we chose the scenario most close to the human operator service, i.e. assuming that the city already has been determined, we start out with the last name. Then, subsequent questions are posed for first and street name. Furthermore, to avoid the problem of the hardly discernible last names (cf. previous subsection), as the entrance step, a complete spelling of the last name is employed.
Even if this scenario is not optimal from the recognition performance's point of view (which would call for starting with the street names), it may be of a greater practical performance as many users will not know the street names of the persons they are asking for. This might become even more applicable if the amount of spelling could be restricted to the first few letters.

Now, Tables 4 – 6 and Figures 5 – 8 show, besides the baseline result of spelling only, the word and graph error rates and the percentage of safe rejections for the different combination methods. Here, the percentage of safe rejections is the number of cases in which the intersection of all combined n-best lists is empty, i.e., in these cases the system knows that it did not understand the user (cf. subsection II.D).
The left column of the tables indicates the size of the databases used. It has to be noted that with increasing level of combination information the number of recognition units increases. Thus, whereas in Table 4 the recognition units consist of the (spelled and spoken) last names only, in Table 6, we deal with all combinations of last, first, and street names. Correspondingly, the lexicon size, i.e. the number of "words" to be recognized increases considerably, e.g. for "100% Berlin" from 189,352 to 1,263,957.
This increase of lexicon size of course counteracts the increase in knowledge gained by the combinations. Together with the pruning effects, this leads for the separate recognition approaches SEP and SEP* even to an increase in word error rate. But also for the hierarchical recognition HIER, one observes an noticeable increase in graph errors thus deteriorating the potentially achievable word error rates. E.g. for "100% Berlin", the GER increases from 15.2% (Tab. 4) to 18.6% (Tab. 6) at an WER of only 19.4% for the full combination (Tab. 6).

From these figures, the following observations can be drawn:

1. There is a substantial pruning problem clearly visible from comparing the results of SEP and SEP* (compare e.g. 21.3% to 48.4% for the spelled/spoken last name combination of "10% of Berlin" in Table 4).
2. Whereas, similar to [7], for method SEP the pure spelling result is always better than the combined recognitions, for SEP* this deterioration is much reduced. Moreover, the effect depends on the size of the database and the amount of combination items and is mainly caused by the graph errors. But, as a consequence, for this recognition setup, pruning errors completely prevent the usage of the additional spoken information.
3. The hierarchical recognition HIER avoids most of the pruning problem and even substantially improves the artificial SEP* results which still suffer from the remaining n-best list pruning effects. This indicates that actually pruning is the only effect deteriorating the SEP results and there are no additional benefits from truncating the n-best lists in the HIER approach. The effect that speaking a name in addition to spelling it can only be advantageously used by the HIER method was already stated in [7]. Unfortunately, details on graph errors are not given there.
4. Relating word and graph error rates to the database size and the amount of combination information it is obvious that with every database the recognition is able to achieve very low error rates as soon as enough knowledge sources are available. I.e. at this

stage the remaining errors are completely determined by graph errors. E.g. for "1% Berlin" one can do with spelled/spoken last plus spoken first name, for "100% Berlin" one additionally needs the street.

5. But even with the hierarchical approach the graph errors increase with each additional knowledge source and they increase even stronger than the rate of safe rejections. Thus, pruning still presents a serious problem.
6. Besides pruning, the high graph errors of the initial spelling step (15.2%) present the main problem. Again, pruning with letter recognition needs to be improved. But there may also be another alternative, i.e. employing the wordlist constraint directly during building-up the spelling graph e.g. via a recognition network. [7] reported very satisfactory results on that.
7. Besides all problems mentioned above, the absolute figures still are very encouraging:[2] Even for the 1.3 million entry database of Berlin we can provide a recognition accuracy of more than 80% accompanied by a rate of safe rejections of about 10%, thus only leaving about 10% of real errors (last row of Tab. 6) which need to be treated in a further dialogue step (and then maybe transferred to a human operator).

Summarizing, the hierarchical recognition setup is a good candidate for completely automating directory assistance for whole countries at a very high accuracy. In addition, this approach is computationally much more efficient as for all but the first recognition step only small lexica have to be employed.

IV. CONCLUSIONS

We showed that with a hierarchical recognition setup satisfactory error rates can be achieved also for very large cities (80% correct, 10% safe rejections). This opens up the possibility for automating directory assistance for complete countries.

Pruning effects causing large graph error rates are identified as the most urging direction for further research. If this problem can be solved the automation of directory assistance is possible without any spelling for medium-size towns (about 100,000) and probably with a minimal amount of spelling (e.g. the first few letters of the last name) even for the largest cities.

[2] Remember the mismatch between training and test for our spoken word recognizer. Latest results show that the recognition rate can be improved through MLLR adaptation by about 25% relative.

% Berlin/ lex. size	method	WER[%]	GER[%]	Rej[%]
0.1 1,824	*spell. only*	*15.7*	*15.2*	*14.2*
	SEP	19.2	18.8	17.6
	SEP*	18.8	18.3	17.2
	HIER	15.8	15.2	14.2
1 10,099	*spell. only*	*17.3*	*15.2*	*12.9*
	SEP	26.2	24.3	21.9
	SEP*	19.8	18.3	16.1
	HIER	17.3	15.2	12.9
10 56,993	*spell. only*	*20.4*	*15.2*	*11.0*
	SEP	48.4	46.5	41.8
	SEP*	21.3	18.5	15.5
	HIER	19.5	15.2	11.0
100 189,352	*spell. only*	*25.6*	*15.2*	*9.3*
	HIER	22.9	15.4	9.5

Tab. 4 – Last name spelled, then spoken: Word Error Rate WER, Graph Error Rate GER, and rejection rate Rej

% Berlin/ lex. size	method	WER[%]	GER[%]	Rej[%]
0.1 1,925	SEP	22.3	22.2	20.6
	SEP*	20.9	20.7	19.4
	HIER	15.5	15.5	14.7
1 13,416	SEP	28.3	27.7	24.7
	SEP*	21.3	20.7	18.5
	HIER	16.1	15.5	13.0
10 123,567	SEP	52.5	50.4	45.0
	SEP*	23.2	20.9	17.6
	HIER	18.5	15.5	11.0
100 961,894	HIER	22.9	16.3	9.5

Tab. 5 – Last name spelled, then spoken, then first name spoken: Word Error Rate WER, Graph Error Rate GER, and rejection rate Rej

% Berlin/ lex. size	method	WER[%]	GER[%]	Rej[%]
0.1 1,934	SEP	27.7	27.4	26.4
	SEP*	24.1	23.8	23.1
	HIER	16.3	16.1	15.2
1 13,451	SEP	35.5	35.2	34.2
	SEP*	24.1	23.8	23.1
	HIER	16.3	16.1	13.6
10 128,608	SEP	55.5	55.2	53.0
	SEP*	24.3	24.0	22.6
	HIER	16.9	16.7	11.7
100 1,263,957	HIER	19.4	18.6	10.2

Tab. 6 – Last name spelled, then spoken, then first name spoken, then street name spoken: Word Error Rate WER, Graph Error Rate GER, and rejection rate Rej

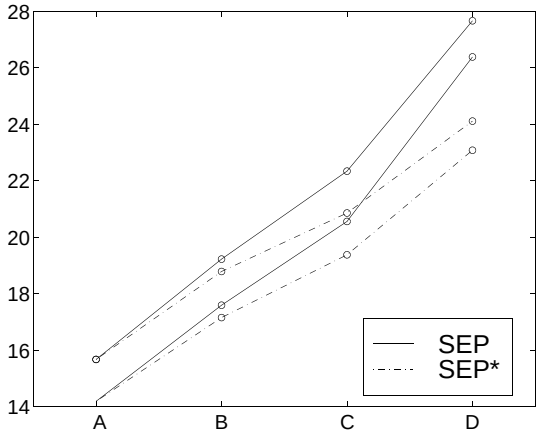

Fig. 5 – Word Error Rates (upper curves) and Rejection Rates (lower curves) versus level of combination for the separate recognitions SEP and SEP* and the "0.1% Berlin" database (1,955 database entries) (A: spelling only, ..., D: +first and + street name)

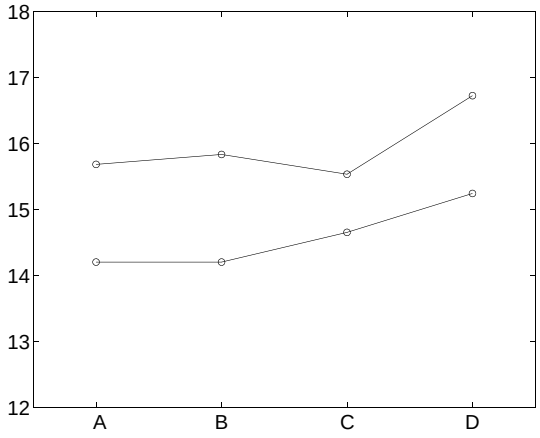

Fig. 6 – Word Error Rates (upper curves) and Rejection Rates (lower curves) versus level of combination for the hierarchical recognition HIER and the "0.1% Berlin" database (1,955 database entries) (A: spelling only, ..., D: +first and + street name)

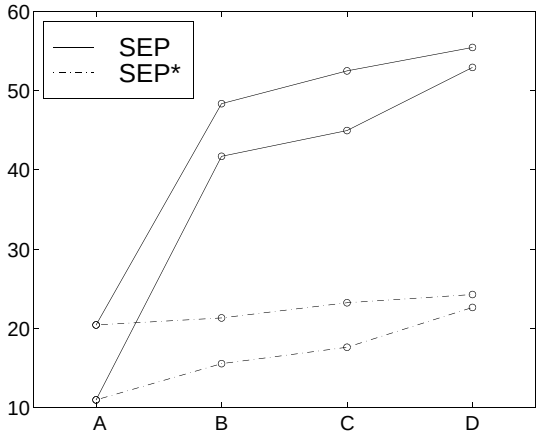

Fig. 7 – Word Error Rates (upper curves) and Rejection Rates (lower curves) versus level of combination for the separate recognitions SEP and SEP* and the "10% Berlin" database (128,642 database entries) (A: spelling only, ..., D: +first and + street name)

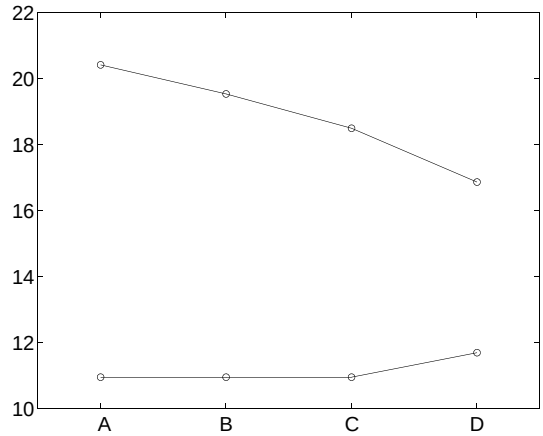

Fig. 8 – Word Error Rates (upper curves) and Rejection Rates (lower curves) versus level of combination for the hierarchical recognition HIER and the "10% Berlin" database (128,642 database entries) (A: spelling only, ..., D: +first and + street name)

REFERENCES

[1] 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, Vigsó, June 1995.

[2] B. Kaspar, G. Fries, K. Schuhmacher, and A. Wirth, "Faust - a directory-assistance demonstrator," in Proc. EUROSPEECH, pp 1161-1164, Madrid, 1995.

[3] F. Seide and A. Kellner, "Towards an automated directory information system," in Proc. EUROSPEECH, vol. 3, pp 1327-1330, Rhodes, 1997.

[4] M. Lennig and G. Bielby, "Directory assistance automation in Bell Canada: Trial results," in Proc. IVTTA, pp 9-13, Kyoto, Japan, 1994.

[5] C. A. Kamm, K.-M. Yang, C. R. Shamieh, and S. Singhal, "Speech recognition issues for directory assistance applications," in Proc. IVTTA, pp. 15-19, Kyoto, Japan, 1994.

[6] H. Hild and A. Waibel, "Recognition of spelled names over the telephone," in Proc. ICSLP, vol. 1, pp 346-349, Philadelphia, 1996.

[7] M. Meyer and H. Hild, "Recognition of spoken and spelled proper names," in Proc. EUROSPEECH, vol. 3, pp 1579-1582, Rhodes, 1997.

[8] http://www.phonetik.uni-muenchen.de/SpeechDat.html

[9] B. Rueber, "Obtaining confidence measures from sentence probabilities," in Proc. EUROSPEECH, vol. 2, pp 739-742, Rhodes, 1997.

[10] H. Aust and M. Oerder, "Dialogue control in automatic inquiry systems," in ESCA Workshop on Spoken Dialogue Systems, pp 121-124, 1995.