INTERNATIONAL JOURNAL OF SPEECH TECHNOLOGY 6, 145–159, 2003

A Speech-Based Human-Computer Interaction System for Automating Directory Assistance Services

K. GEORGILA AND K. SGARBAS
Wire Communications Laboratory, Electrical and Computer Engineering Department, University of Patras, Greece
rgeorgil@wcl.ee.upatras.gr

A. TSOPANOGLOU
Knowledge S.A., LogicDIS Group, Patras, Greece

N. FAKOTAKIS AND G. KOKKINAKIS
Wire Communications Laboratory, Electrical and Computer Engineering Department, University of Patras, Greece

Abstract. The automation of Directory Assistance Services (DAS) through speech is one of the most difficult and demanding applications of human-computer interaction because it deals with very large vocabulary recognition issues. In this paper, we present a spoken dialogue system for automating DAS.[1] Taking into account the major difficulties of this endeavor a stepwise approach was adopted. In particular, two prototypes D1.1 (basic approach) and D1.2 (improved version) were developed successively. The results of D1.1 evaluation were used to refine D1.1 and gradually led to D1.2 that was also improved using a feedback approach. Furthermore, the system was extended and optimized so that it can be utilized in real-world conditions. We describe the general architecture and the three stages of the system's development in detail. Evaluation results concerning both the speech recognizer's accuracy and the overall system's performance are provided for all prototypes. Finally, we focus on techniques that handle large vocabulary recognition issues. The use of Directed Acyclic Word Graphs (DAWGs) and context-dependent phonological rules resulted in search space reduction and therefore in faster response, and also in improved accuracy.

Keywords: speech-based human-computer interaction, automatic directory assistance services (DAS), telephone-based, large vocabulary speech recognition, directed acyclic word graphs (DAWGs), context-dependent phonological rules

1. Introduction

Automatic inquiry systems are systems that people can call in order to obtain certain information, without a service representative being involved. The system has to create a database query from the user's input and present the results to him/her. In the first attempts for automation, callers had to interact with automatic inquiry systems by pushing keys on their touch-tone telephone. The ensuing dialogue was usually menu-driven, rigidly structured, and accompanied by lengthy explanations (Aust et al., 1995). A promising solution to this problem was the use of speech. Thus, the introduction of speech recognition for digits and small vocabularies led to the first successful commercially available spoken dialogue systems. Nevertheless, the use of such systems is still limited to simple applications due to their system-driven and menu-style dialogues, and their small recognition vocabulary. As soon as the application gets more complex, the human-machine menu-based interaction becomes very lengthy and monotonous, and consequently hardly acceptable by users (Gardner-Bonneau, 1992).

In order to overcome the aforementioned limitations, major efforts have been undertaken in the last decade to develop systems with larger vocabularies and more user-friendly, mixed-initiative dialogues based on speech understanding. The most typical applications of spoken dialogue systems involve making travel arrangements (Aust et al., 1995; Glass et al., 1995; Lamel et al., 2000), inquiring about weather (Zue et al., 1997), telephone banking (Sugamura et al., 1998), requesting insurance transactions (Georgila et al., 1998), call-routing (Gorin et al., 1997), conference services (Rahim et al., 2001), restaurant guides (Jurafsky et al., 1994), and finally Directory Assistance Services (DAS).

The automation of DAS is one of the most difficult and demanding applications of speech recognition, which supersedes other large vocabulary applications in terms of complexity and vocabulary size. It has attracted great interest in the last decade due to the visible benefits both for the telephone companies and the subscribers (Lennig et al., 1995). Several prototypes have been reported, such as the system of British Telecom (Whittaker and Attwater, 1995), PADIS-XL (Seide and Kellner, 1997), the Durham telephone enquiry system (Collingham et al., 1997), and ADAS Plus (Gupta et al., 1998). Other tasks relevant to DAS are the automation of collect and third-party-billed calls (Lennig, 1990) and automatic name dialing (Gao et al., 2001).

In this paper, we present a speech-based human-computer interaction system for automating DAS that was developed in the framework of the EU project IDAS (Interactive telephone-based Directory Assistance Services), and then extended and improved so that it can be utilized in real-world conditions. Another prototype also funded by IDAS has been reported in Córdoba et al. (2001). The primary target of IDAS was to demonstrate the applicability of very large vocabulary speech recognition and spoken dialogue technologies in the development of cost-effective and user-friendly applications for automated (without the intervention of human operators) and interactive telephone-based DAS. The project was carried out by 10 partners from Germany, Greece, Spain, and Switzerland. In this paper, we will describe the Greek dialogue system developed for IDAS.

Taking into account the major difficulties of this endeavor a stepwise approach was adopted. In particular, two prototypes D1.1 (basic approach) and D1.2 (improved version) were developed successively. The results of D1.1 evaluation were used to refine D1.1 and gradually led to D1.2. D1.2 was also verified and improved using a feedback approach. Finally, the system was extended and optimized to be used in a real environment.

The paper is organized as follows: Section 2 presents the architecture of the system. The Greek prototypes D1.1 and D1.2 are described in Sections 3 and 4 respectively. The extended final version is presented in Section 5, whereas a summary and some conclusions together with ideas for future work follow in Section 6.

2. System Architecture

The system consists of the following modules: system control, switch, dialogue, speech input, speech output, database, operator and Graphical User Interface (GUI). Each module contains one or more components. A component is an encapsulated piece of software that offers a clearly defined functionality. Figure 1 shows the system's modules and their interaction while Fig. 2 depicts the components of which the above modules are composed.

The *System Control* manages the data transfer among modules, except for the audio signal data, which are directly passed between the line interface and the speech recognizer and synthesizer. The *Switch* and *Line Interface* handle incoming calls and switching functionality. Furthermore, they detect if a client has hung up during the dialogue and notify the system control (Hennecke et al., 1999).

The *Dialogue Manager* is responsible for the dialogue flow, keeping record of the history and the information retrieved from the database. It also returns a set of recognizer parameters (e.g., sub-vocabulary and/or language models) that will be activated in the next recognition step, and generates the system prompts that will be passed to the speech synthesis component. In addition, the dialogue manager provides parameters to control the behavior of both speech input and speech output (e.g., timeout, interruptions). In order to determine how to continue the dialogue manager may need data from some external database. In this case, the dialogue manager sends back a database request to the system control that instructs the *Database Manager* to perform the request. The result of the database lookup

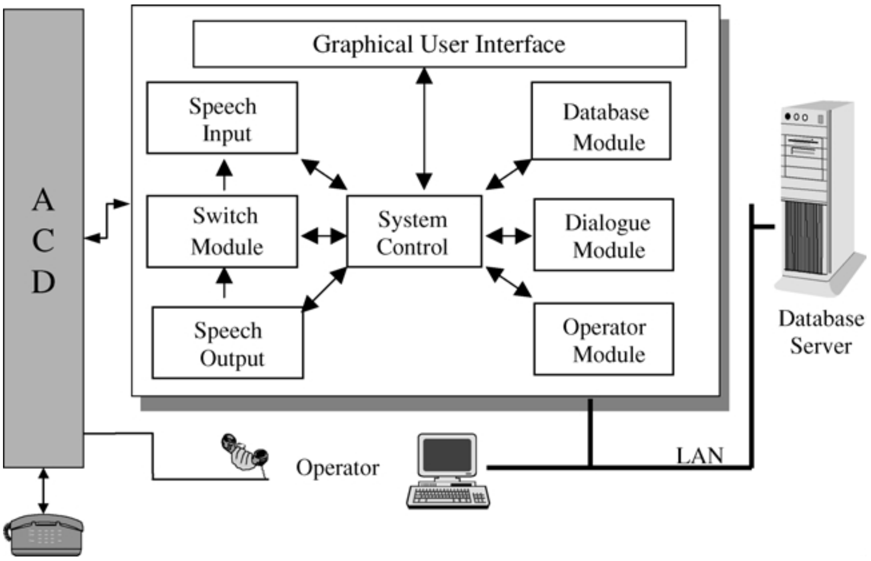


Figure 1. System architecture.

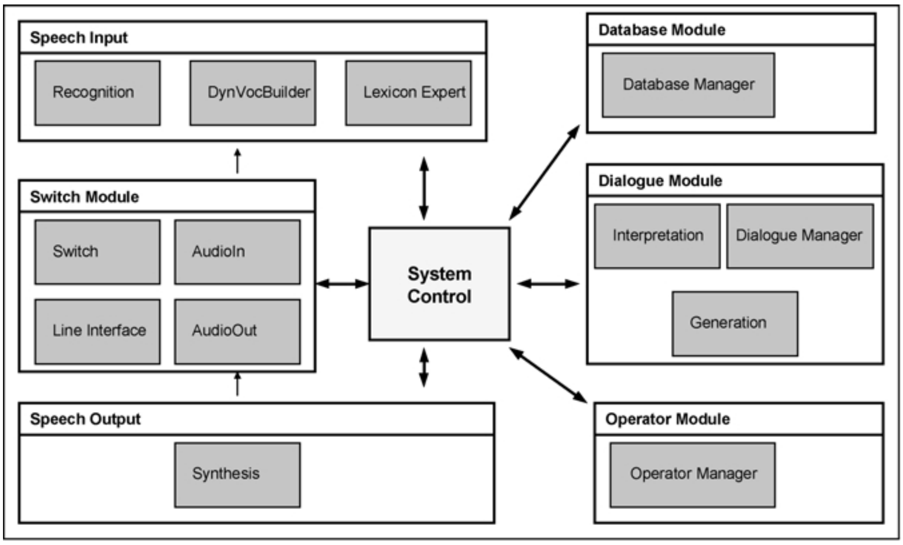


Figure 2. Decomposition of modules.

together with the request for the next dialogue step is returned to the dialogue manager by the system control.

In prototypes D1.1 and D1.2, the *Speech Recognizer* we used was built with the HTK Hidden Markov Models toolkit (Young et al., 1997). However, in the final extended version, the Philips SpeechPearl 2000 recognition engine was incorporated into the system. The acoustic models are tied-state context-dependent triphones of five states each. In order to train the recognizer we used the SpeechDat-II Greek telephone database (Van den Heuvel et al., 2001). This database is a collection of Greek annotated speech data from 5000 speakers (each individual having a 12-minute session). We made use of utterances taken from 3000 speakers in order to train our system. Each input speech signal waveform is sampled at 8 kHz, pre-emphasized by the filter $H(z) = 1 - 0.97z^{-1}$ and subsequently windowed into frames of 20 ms duration at a frame rate of 10 ms using a Hamming window. Thirteen-dimension feature vectors are formed, that is, 12-dimension Mel Frequency Cepstral Coefficients plus a log-energy value. Cepstral mean normalization is applied to deal with the linear channel assumption. The 13 aforementioned coefficients and their temporal regression coefficients of first and second order form the final 39-dimension observation vector.

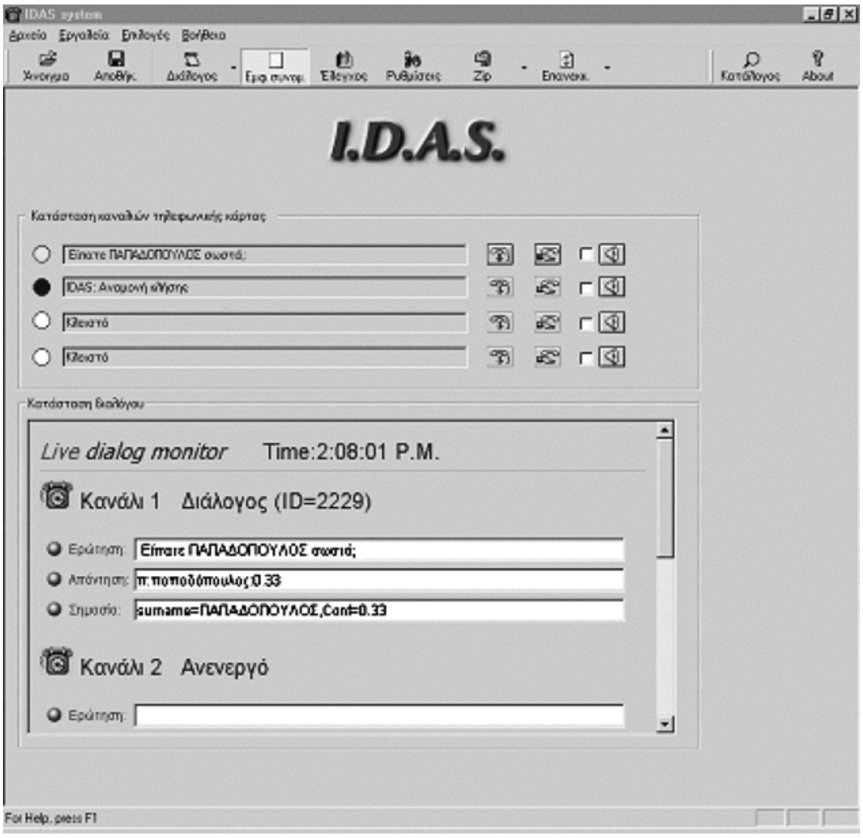
We use dialogue state dependent language models formed as lattices. In prototypes D1.1 and D1.2 grammar-based language models were applied. In the new extended version, we experiment with bigrams and a novel method we have developed, which produces stochastic finite-state networks that incorporate grammatical structure provided by large-context dependencies as well as coverage of ungrammatical spontaneous sentences provided by statistical estimations (Georgila et al., 2001a).

The recognition result is passed to the *Semantic Interpretation* component in which linguistic analysis and contextual interpretation is carried out. *Speech Synthesis* is accomplished by using a mixture of prerecorded speech (for prompts) and synthesized speech (for surnames, letters, and digits) (Gong and Lai, 2001).

In those cases in which the dialogue task cannot be completed by the system, the control is transferred to the human operator. The transfer of data to the human operator is handled by the *Operator Manager*, which in turn relies on the switching functionality of the switch module to handle the transfer. If the human operator is not available, the operator manager will return the appropriate exception status and the system control will ask the dialogue manager to provide a suitable dialogue step for continuation, e.g., say goodbye to the caller.

A barge-in capability is supported (in D1.2 and the final extended version), that is, the user may interrupt the system and speak before the system prompt is completed. Echo cancellation, applied to the recorded signal, is used to remove the echo of the synthetic speech so that the system is able to detect if the caller is speaking. When speech is detected, synthesis is stopped.

Figure 3 depicts the GUI module that is responsible for system configuration and starting or shutting



○ Είπατε ΠΑΠΑΔΟΠΟΥΛΟΣ σωστά;

● IDAS: Αναμονή κλήσης

○ Κλειστό

○ Κλειστό

Live dialog monitor Time:2:08:01 P.M.

Κανάλι 1 Διάλογος (ID=2229)

Ερώτηση: Είπατε ΠΑΠΑΔΟΠΟΥΛΟΣ σωστά;

Απάντηση: π:ποποδόπουλος:0.33

Σημασία: surname=ΠΑΠΑΔΟΠΟΥΛΟΣ,Conf=0.33

Κανάλι 2 Ανενεργό

Ερώτηση:

Figure 3. The graphical user interface (GUI) module.

the system down. In addition, it provides visualization of the operation status (e.g., tracing, indicating errors). The role of the *Dynamic Vocabulary Builder* and the *Lexicon Expert* will be explained in the following sections.

The dialogue system runs on a Pentium II or III PC with at least 400 MHz (preferably Dual), 6 GBytes hard disk, and 256 MBytes RAM. It is connected to the public telephone network or to a PBX via the Dialogic D41ESC card. The input signal is encoded in A-law, 8 bit, 8 kHz speech signal.

SPRADIAK (Kaspar et al., 1997) was a first approach (D0) towards a partial automation of DAS in Germany. SPRADIAK prompts the user for information in a strict dialogue. It recognizes the call category (private, business, etc.) and up to 50 city names (frequently asked at the test site). Other user inputs (surname, first name, etc.) are recorded. Recognition results are automatically shown on the operator screen. The operator can then simultaneously listen to the recorded utterances in a formatted way, and fill in the rest of the information needed to start a database query.

Based on the experience from D0 and the analysis of human-human dialogues, the functional and the design specifications of D1.1 and D1.2 were established. The development and verification stages were characterized by the stepwise enhancement of the system's functionality. The results of D1.1 evaluation with limited users at the telephone company's site were used to refine D1.1 and gradually increase the vocabulary till D1.2 was achieved. D1.2 was also verified and improved using a feedback approach. Finally, the system was extended and optimized so that it can be utilized in real-world conditions. That is, real-time response of the recognizer was achieved and its accuracy increased. Moreover, in order to decrease the average duration and number of turns, the confirmation dialogue states were discarded and empirically set confidence levels were applied. In the following, the Greek prototypes D1.1 and D1.2 and the final improved and extended version of the system are presented in detail.

3. Prototype D1.1

In the first step of the dialogue the system asks the user if s/he is looking for the telephone number of a company, an organization/institute or a person. A typical dialogue in which the caller requests a person's telephone number is as follows:

. . .
System: *Have you called for the phone number of a company, an organization or institute, or a person?*
User: *Of a person.*
System: *Did you say person?*
User: *Yes.*
System: *Please give the city name.*
User: *Athens.*
System: *Did you say Athens?*
User: *Exactly.*
System: *Could you please specify the district?*
User: *I think it is Marousi.*
System: *Did you say Marousi?*
User: *Yes, Marousi.*
System: *Please give the person's surname.*
User: *He is called Varelis.*
System: *Did you say Varlis?*
User: *No, I said Varelis.*
System: *Please repeat the person's surname.*
User: *Varelis.*
System: *Did you say Varelis?*
User: *That's right.*
System: *Please give the forename of the person.*
User: *Stefanos.*
System: *Did you say Stefanos?*
User: *Yes.*
System: *The number you requested is . . .*

After the system has gathered the necessary information, it searches the telephone directory, and the telephone number asked for is spoken to the user as a mixture of prerecorded speech (for the prompt) and synthesized speech (for the digits that form the telephone number). If the search in the database produces more than one solution, the system will inform the user about all of them. The system repeats what it heard to the caller and asks for confirmation. If the user does not confirm the recognizer's output, then s/he is prompted for repetition. The confirmation dialogue state is activated again and three successive negative confirmations entail the call's transfer to a human operator along with the already gathered information.

If the user gives the city name of Athens or Thessaloniki (the biggest cities in Greece), the system will prompt him/her to specify a district in the above city. However, the caller could also give directly the name of the district, without having to utter the city name first. In those cases in which the system cannot find the requested telephone number in the district provided by the caller, it will extend the search space to the other districts of the city as well. Thus, it is ensured that even if the user has no knowledge about the exact district, which happens very often, s/he will be able to get the desired information.

No barge-in is included in this prototype. The vocabulary of the recognizer includes the 10,200 most frequent surnames, 326 first names, 342 cities, 200 neighborhoods, and 50 additional words. The vocabulary differs from node to node with perplexity reaching its maximum value at the node recognizing the surname and its minimum value at the call type node in which the system asks for the kind of the inquiry (person, organization/institute, company, etc.). We should stress here the fact that the user is not restricted in any dialogue state to uttering isolated words but s/he is allowed to give complete utterances.

The speaker-independent continuous spontaneous speech recognizer was built with the HTK Hidden Markov Models toolkit (Young et al., 1997). It was tested with approximately 100 words (24 letters, 10 digits, 65 words) by 30 speakers (18 male, 12 female, students and employees at the University of Patras and Knowledge S.A.) with 36 items per speaker. The HTK HResults tool was used to assess the recognizer's accuracy. Results measured at word and sentence level are given in Table 1. The following terminology is used to interpret the results: H stands for the number of hits (correct identification) and S for the number of substitutions. D and I correspond to the number of deletions and insertions respectively. Finally N is the total number of words or sentences.

Accuracy is given at word and sentence level. For sentences the accuracy is defined as $\%\text{Corr} = \frac{H}{N} \times 100$, where $N = H + S$. There are two metrics for words: The first is defined as $\%\text{Corr} = \frac{H}{N} \times 100$ and gives the percentage of hits, while the second is defined as $\%\text{Acc} = \frac{H-I}{N} \times 100$ and gives the number of hits taking into account inserted words as well. It should be noted that at the word level $N = H + D + S$.

One of the most critical factors that influence the performance of the recognizer is the background noise level. The mean noise level of the testing environment was 71.1 dB, and the maximum value was 96.7 dB. These noise levels are much higher than the predicted ones for the laboratory environment (40–50 dB) and therefore the given performance could have been influenced negatively.

Table 1. Accuracy of D1.1 recognizer.

	Correct (%)	Acc (%)	H	S	D	I	N
Word	93.10	91.62	5,656	372	47	90	6,075
Sentence	59.01	–	429	298	–	–	727

H: Hits (correct), S: Substitutions, D: Deletions, I: Insertions, N: Total number of words or sentences.

Table 2. Average number of system-user exchanges (D1.1 prototype).

Node	Number of system-user exchanges
Greeting	1
Call type	1.05
City	1.43
Surname	1.65
First name	1.07
Tel_response	1
Thanks	1
Operator	1
Neighborhood	1.01
Total number	10.21

In contrast with the glass-box evaluation of the speech recognizer the system control and the dialogue module were tested by the same users that followed real scenarios, making real-world transactions. Table 2 gives the average number of system-user exchanges. The task completion rate is 68.92% if the empty recordings (the user does not respond to the system prompt) are used for the estimation of the completion rate. In the case that the empty recordings are not included, a completion rate of 84% is achieved. The transaction completion rate is presented in Table 3. The transaction completion rate, considering two trials, has been measured to 100% for the recognition of the call type, 86%

Table 3. Transaction completion rate (D1.1 prototype).

Node	1st trial (%)	2nd trial (%)	3rd trial (%)	4th trial (%)	Not recognized (%)
Greeting	0	0	0	0	0
Call type	90	10	0	0	0
City	62	24	9.5	3.8	0.7
Surname	39	26.8	7.3	12.2	14.7
First name	85.4	9.8	2.4	2.4	0
Tel_response	0	0	0	0	0
Thanks	0	0	0	0	0
Operator	0	0	0	0	0
Neighborhood	85	10	4.5	0.5	0

Table 4. Mean duration per interaction (D1.1 prototype).

Node	System prompt duration (sec)	User answer duration (sec)	Total turn duration (sec)
Greeting	4.33	0	4.33
Call type	3.86	3.43	7.29
City	3.68	3.51	7.19
Surname	2.68	3.10	5.78
First name	2.65	3.14	5.79
Tel_response	4.00	0	4.00
Thanks	2.13	0	2.13
Operator	3.74	0	3.74
Neighborhood	2.60	3.00	5.60
Mean duration	3.30	1.79	5.09
Total duration	29.67	16.18	45.85

for city names, 65.8% for surnames, 95.2% for first names and 95% for neighborhood names. Table 4 depicts the mean duration per interaction. The mean turn duration is 5.09 sec and the mean transaction duration is 45.85 sec. The conclusion from the above given results is that a usual transaction takes rather long time (almost 46 sec) and the greatest part of this time is consumed by the system. Another thing that must be taken into account is that all system nodes contain confirmation steps. Considering that almost 1.5 sec of silence (depending on the line interface card settings) in every user input is added to the duration of the user phrase, we can estimate that the net duration of user phrases is 0.29 sec that is 1.79 sec–1.5 sec or 300 msec per node in average.

In order to assess the users' satisfaction each user was asked to fill in a questionnaire with four questions at the end of the evaluation procedure. The four questions were:

Question 1: Did you know what you should do before using the system?

Yes □ Partly □
No □ No answer □

Question 2: Did you manage to accomplish your intention using the system?

Yes □ Partly □
No □ No answer □

Question 3: Are you satisfied with the system's answers?

Yes □ Partly □
No □ No answer □

Table 5. Questionnaire results (D1.1 prototype).

Question/ Answer	Yes (%)	Partly (%)	No (%)	No answer (%)
1	80	5	15	0
2	15	80	0	5
3	15	70	10	5
4	20	55	20	5
Mean value	32.5	52.5	11.25	3.75

Table 6. Mean user correction rate (D1.1 prototype).

Node	User correction rate (%)
Greeting	0
Call type	5
City	43
Surname	65
First name	7
Tel_response	0
Thanks	0
Operator	0
Neighborhood	1
Mean rate	13.44/24.2

Question 4: Is the system good (would you use it)?

Yes □ Partly □
No □ No answer □

The results of the above procedure are shown in Table 5. We conclude that the majority of the users (85% positive answers) were found to be fully or partly satisfied from the system and only 11.25% of the answers were actually negative.

The mean user correction rate (Table 6) is almost 13.5% given that we use all the nodes to estimate the final results. In the case that we take into account only the 5 nodes during which there is a real system-user interaction, the correction rate is increased to 24.2%.

4. Prototype D1.2

Based on the experience gathered from the testing of D1.1, the prototype D1.2 was designed. In D1.2 the recognizer must distinguish among 257,198 distinct surnames that correspond to 5,303,441 entries in the directory of the Greek Telephone Company. By restricting the search space to the most frequent 88,000 ones that correspond to about 123,313 distinct pronunciations, 93.57% of the directory's listings is covered. All city names, first names and neighborhood names were already included in the vocabulary of D1.1. Apart from the vocabulary extension, another very important goal was to make the system more user friendly by allowing barge-in, decreasing the mean transaction time, increasing the dialogue completion rate and ensuring smooth continuation of the dialogue by a human operator in those cases in which the system fails to complete it. Kamm et al. (1995) performed a study on the relationship between recognition accuracy and directory size for complete name recognition and reached the conclusion that accuracy decreases linearly with logarithmic increases in directory size. The above conclusion shows that it is necessary to apply techniques for handling large vocabulary recognition issues.

An efficient search through a large vocabulary structure may be performed by two common methods: the first is to reduce the size of the active vocabulary in every dialogue turn and the second to use spelling.

In our system each dialogue turn is independent of the previous ones. Therefore the search space is not reduced with every dialogue turn. The fact that we have decided to keep dialogue turns independent of each other is a point at which our approach varies from the ones used by Philips and British Telecom in their directory information systems. In the Philips Automatic Directory Information System (Seide and Kellner, 1997), the search space is reduced with every dialogue turn. Note that here the caller must utter only one word per dialogue turn, e.g., "Aachen", whereas in our system there is no such restriction. That is, the utterance "he lives in Athens" is allowed and will be correctly processed. During the development of their DAS system (Whittaker and Attwater, 1995), British Telecom experimented with all sorts of dependencies and reached the conclusion that if recognitions stay independent of each other and the N-best lists are intersected with the database, confidence increases while accuracy drops. In this case the recognition task is more difficult because the entire vocabulary is active. Therefore, if the recognizer provides a solution with high probability then the recognition result is almost certain to be correct, which implies a high value of confidence. On the other hand, if successive recognitions are constrained by previous ones then the recognition task is easier since the active vocabulary is restricted. Thus, accuracy gets higher and confidence decreases.

The reason we have decided to keep dialogue turns independent of each other is that we are interested in high confidence. Nevertheless, experimentation with constrained recognitions by previous ones is a process in progress, which requires that the speech recognizer be improved so that possible recognition errors do not affect the subsequent dialogue turns. An additional reason for the independence of dialogue turns is that it deals with the problem that would arise otherwise if the caller gave a false district. If the search space was reduced with every dialogue turn and the system failed to find the requested information in the district specified by the user, it would not have the alternative solution of extending the search to other districts in the same city. This is because the list of active surnames or first names would have been limited to include only surnames and first names of the selected district. The above mentioned systems of Philips and British Telecom use spelling to restrict the search space. In Greek, spelling is not usual (splitting the word in syllables is preferred), and thus we have decided not to use it in our dialogue system.

Our decision not to reduce the search space with every dialogue turn entails that in the case of surname recognition, the recognizer will have to perform an extremely difficult task. That is, to find the correct surname (the one uttered by the user) among the 88,000 most frequent ones (123,313 distinct pronunciations). To make things easier, the system prompts the caller to give the first three letters of the surname and then fully utter it. The dialogue of the D1.1 prototype given in Section 3 is going to become as follows:

...

System: *Please utter the first three letters of the person's surname.*
User: *V A R.*
System: *Did you say V A R?*
User: *Yes, this is what I said.*
System: *Please give the person's surname.*
User: *He is called Varelis.*

...

However, if the caller has chosen to ask for the telephone number of a company or an organization/institute then there is no need to ask for the first three letters of the company's or organization/institute's name since the total number of existing companies and organizations/institutes is much smaller than the distinct surnames included in the database of the Greek Telephone Company. A typical dialogue in which the caller requests the telephone number of a company or organization/institute is as follows:

. . .

System: *Have you called for the phone number of a company, an organization or institute, or a person?*
User: *Company.*
System: *Did you say company?*
User: *Right.*
System: *Please give the city name.*
User: *The city is Patras.*
System: *Did you say Patras?*
User: *Yes, Patras.*
System: *Please give the name of the company.*
User: *It's Knowledge.*
System: *Did you say Knowledge S.A.?*
User: *Yes, correct.*
System: *The number you requested is . . .*

The speech recognizer is configured dynamically to search only among the surnames that start with the previously recognized three letters. This function is performed by the *Dynamic Vocabulary Builder* component, which is included in the speech input module. The dynamic vocabulary builder is also activated when the user utters the city name of Athens or Thessaloniki and the recognizer searches for the districts of the selected city (also in prototype D1.1). Then the dynamic vocabulary builder restricts the active vocabulary to the districts of the specific city.

Although the active list of surnames is reduced to the ones starting with the previously given three letters, the number of distinct surnames remains high. In order to deal with this issue and ensure real-time speech recognition, we replace the word networks of surnames with phoneme networks that can produce the phonetic transcriptions of all the above surnames. We build the phoneme networks as acyclic (no loops are allowed) finite-state automata—also known as DAWGs (Directed Acyclic Word Graphs)—for two reasons: (a) they allow sharing phones across different words (as opposed to using a separate instance for every phone in the pronunciation of each word), thus reducing recognition search space and response time, and (b) incremental algorithms are currently available (Aoe et al., 1993; Daciuk et al., 2000; Sgarbas et al., 1995, 2001) for the construction of minimal DAWGs thus making the process of lexicon update much more effective and time-efficient.

It is a common practice in most very large vocabulary speech recognition systems to store their lexicons in tree structures (i.e., tries). However, while trees exploit efficiently common word prefixes, they fail to do the same with common word suffixes. For this reason, the use of DAWG structures is more appropriate in this case. DAWGs have been successfully used for storing large vocabularies in speech recognition. Hanazawa et al. (1997) used an incremental method (Aoe et al., 1993) to generate deterministic DAWGs. The aforementioned method was applied on a 4000-word vocabulary in a telephone directory assistance system. However, in Hanazawa et al. (1997) the tree and the DAWG performances were not measured under the same conditions. Different decoding algorithms were used for each implementation, so the comparison results were not completely accurate. Betz and Hild (1995) used a minimal graph to constrain the search space of a spelled-letter recognizer. However, neither did they report details on the algorithm they applied, nor did they compare the performance of graphs against full-forms (i.e., whole words with no merging of nodes) and trees. In Georgila et al. (2000), DAWGs are compared against full-forms and trees for different vocabulary sizes and pruning levels under the same conditions and using the same decoder (HTK decoder), thus providing comparable results indicating that DAWGs supersede full-forms and trees in terms of compactness (i.e., memory size) and response time.

In the present work, DAWG-based lexicons have been incorporated in a real-world application. We have used incremental construction algorithms (Sgarbas et al., 1995, 2001) in order to update non-deterministic DAWGs as frequently as required, without having to rebuild them from scratch every time. We are particularly interested in non-deterministic DAWGs because they require even less space than the corresponding minimal deterministic ones (Sgarbas et al., 2001).

The whole process can be described as follows: A word (full-form) network consisting of surnames is replaced by a phoneme network that can produce the phonetic transcriptions of all the above surnames (Fig. 4(a) and (d)). Thus, a lexicon of surnames in phonetic transcription (Fig. 4(a)), is first transformed into a DAWG (Fig. 4(c)), where simple monophone pronunciations label the transitions between nodes. The next stage of the method is to convert these structures into the format accepted by the HTK decoder, where the labels are on the nodes (Fig. 4(d)). The corresponding tree structure is given in Fig. 4(b) for comparison reasons. If the surnames in Fig. 4 had multiple pronunciations, they would be treated as different words by the algorithm. Using the above network reduction method, we

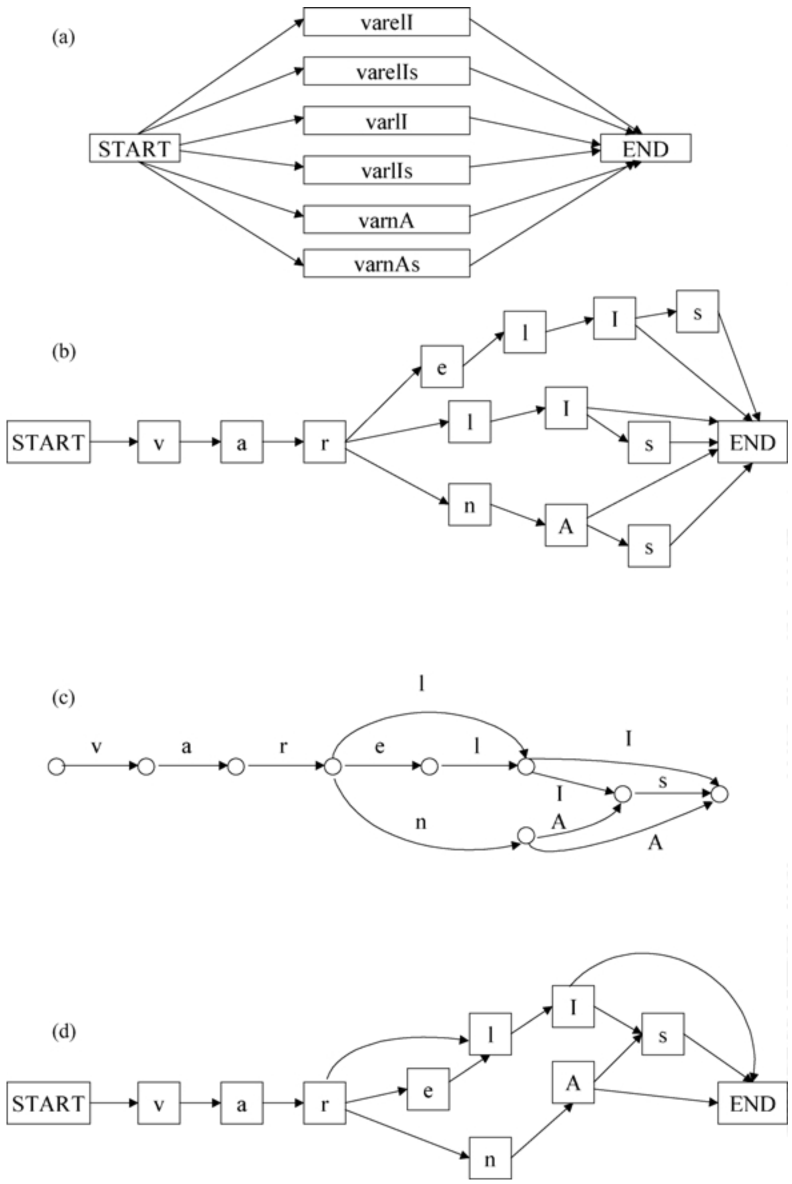


Figure 4. (a) Full-form word network, (b) phoneme tree in the decoder format, (c) phoneme DAWG produced by our incremental algorithm, and (d) phoneme graph in the decoder format.

get an equivalent but more compact network, which results in considerably faster search. In both the tree and the graph several words share common paths, thus recognition is substantially accelerated in comparison to the full-form network, when the same recognizer is used in all networks. Furthermore the graph is more compact than the tree since common suffixes are also merged. The recognition accuracy is retained since the same phoneme combinations are involved. Details are given in Georgila et al. (2000).

The system was tested in 3 phases by students and personnel of the University of Patras, Knowledge S.A. and the Greek Telephone Company. The recognition results are given in Table 7. By that time there was also improvement in the acoustic models, which led to better recognition rates compared to the ones we had during the D1.1 evaluation.

Field tests were carried out with 110 people to evaluate the performance of the automatic directory information system as a whole. The 76 males called

Table 7. Recognition accuracy (D1.2 prototype).

	1st phase (%)	2nd phase (%)	3rd phase (%)
Call type	100.00	100.00	100.00
City	95.50	93.50	89.60
Surname	73.10	68.60	63.00
First name	97.60	94.20	91.10

Table 8. Field tests (D1.2 prototype).

	Recognition accuracy (%)	Number of turns	User correction rate (%)
Call type	98.77	1.012	1.24
City	81.14	2.372	16.15
First 3 letters	89.65	2.200	6.92
Surname	70.85	2.330	14.57
First name	93.34	2.188	7.25
Totals		10.102	

the system 381 times, and the 34 females 123 times. These people were chosen to cover different ages, dialects and education levels. The results are presented in Table 8. The recognition accuracy was 98.77% for the subtask of call type (company, organization/institute or person), 81.14% for city names, 89.65% for giving the first three letters of the surname, 70.85% for surnames, and 93.34% for first names.

If we consider only the nodes where the user interacts with the system, the average dialogue duration is 69.13 sec, which is greater than the duration we had in D1.1 (45.85 sec). This is justified by the fact that there is an additional node for the first three letters of the surname. Moreover, the turn duration in the node at which the caller gives the surname is increased due to the large vocabulary size. If we also consider the duration of the system prompts and the database search the average duration rises to 110.76 sec. Table 9 shows that the average number of turns is 15.227.

Confirmation is used in every dialogue state in the same way as in the D1.1 prototype. Thus, when the system asks the user to confirm whether the recognized surname is correct or not, the speech synthesizer is used to speak out the recognized surname. However, the low quality of speech synthesis causes problems. Table 10 depicts the call analysis. The number of calls serviced by the system was 315 (62.50%), while 84 calls (16.67%) were forwarded to a human operator and 105 (20.83%) were missed. Table 11 shows the analysis of the missed calls. Fifty one of them (10.12% of the total number of calls, synthesizer X) arose from the fact that the recognizer recognized correctly the uttered surname, but when the synthesizer pronounced this name and asked for confirmation, the user did not understand that the name uttered was the correct one and gave a negative confirmation. On the other hand, ten calls (1.98% of the total number of calls, synthesizer Y) were missed because the recognizer produced an invalid surname. This surname was uttered by the synthesizer, the user thought that the correct name was pronounced, and gave a positive confirmation. In Table 10 we can see that there were 22 problematic calls, which resulted from connection errors (13) or problems in the user-system interaction (9). The remaining 105 problematic calls were missed calls.

Table 9. Average number of turns (D1.2 prototype).

	Average number of turns
Greeting	1
Dialogue	10.102
Prompt and database search	1
Info	3.125
Total	15.227

Table 10. Analysis of calls (D1.2 prototype).

	Percentage (%)	Totals	Problems	
			No	Yes
Serviced by system	62.50	315	315	0
Serviced by human operator	16.67	84	62	22
Missed calls	20.83	105	–	105
Totals		504	377	127

Table 11. Analysis of missed calls (D1.2 prototype).

	Number of calls	Percentage of total number of calls (%)
Synthesizer X	51	10.12
Synthesizer Y	10	1.98
Database	8	1.59
Dialogue	12	2.38
User	24	4.76

5. Final Extended Version

Since there is no dialogue turn for spelling and the caller is prompted directly to utter the surname, the value of N in the N-best hypotheses' list of the speech recognizer must be high. This will ensure that the correct surname (the one uttered by the user) is included. There are many acoustically similar surnames, and if N is small it is very likely that the correct surname does not appear in the list because the N positions of the list are all occupied by surnames acoustically similar to the correct surname. However, a very high value of N will slow down the system's response.

In order to cope with the above problem, context-dependent phonological rules are applied to the N-best hypotheses produced by the speech recognizer. These rules define classes of phonemes and phoneme combinations, the members of which can be falsely recognized in a specific context. That is, a phoneme or phoneme combination of a class could be mistaken for another phoneme or phoneme combination of the same class in the context defined by the rule. Thus, recognition errors and pronunciation variability are taken into consideration. The solutions created by applying the phonological rules are surnames acoustically similar to the N-best hypotheses produced by the speech recognizer. The rules are language-dependent and they are carefully selected so that they cover the most probable interchanges between phonemes or phoneme combinations, but without leading to too many solutions. On the other hand, the rules' processing algorithm is language-independent.

After the rules are applied, the phonetic transcriptions of the N-best surnames are transformed to the corresponding graphemic ones. Note that a single phonetic transcription could lead to multiple graphemic ones. The above transformation is done automatically since both the phonetic and graphemic transcriptions of a surname are stored in the lexical database. Otherwise a phoneme-to-grapheme converter would be used. The *Lexicon Expert* is responsible both for performing the task of applying phonological rules and for the correspondence of phonetic and graphemic transcriptions. However, in automatic name dialing (Gao et al., 2001), a task similar to DAS, the user may need to add words to his/her personalized vocabulary. Therefore a phoneme-to-grapheme converter or the contrary would not be sufficient. We would have to use algorithms for the automatic generation of pronunciations based on acoustic information (Ramabhadran et al., 1998). In our application, the directory of the Greek Telephone Company cannot be modified by the user and thus such algorithms are not necessary.

Most approaches incorporate pronunciation variation into the lexicon that will be used by the recognizer in the decoding process (Schmid et al., 1993; Ramabhadran et al., 1998). Our proposal is to apply information on pronunciation variation in a separate stage after the recognition task. That is, we apply phonological rules to the recognizer's output. The advantage of such an approach is the gain in response time. The cost of processing the signal in order to produce multiple outputs is much higher than the time required for taking an output and applying the phonological rules.

The structure of the rules is as follows:

$$L_1, L_2, \ldots, L_k, S, R_1, R_2, \ldots, R_n$$

where $L_i\ i = 1, \ldots, k$ is the left context of the rule, S is the class, which includes phonemes or phoneme combinations that could be interchanged, and $R_p\, p = 1, \ldots, n$ is the right context of the rule. The values of k and n could vary according to the language and the way the designer of the rules has decided to form them. Each L_i or R_p is a class of phonemes or phoneme combinations that could substitute one another as context of the central part of the rule S. There are three types of rules (substitution, insertion, and deletion), which contain both phonetic and linguistic knowledge.

Currently the rules are extracted manually. However, research in developing an algorithm for their automatic extraction, is in progress. We aim at developing an algorithm for the automatic extraction of rules, which will exploit both the linguistic knowledge contained in phonetic transcriptions of words, and the information carried in the speech signal itself. Details on the structure of the rules and the way they are processed are given in Georgila et al. (2001b).

In order to evaluate the recognition performance after the application of phonological rules, new tests were carried out. Thus, 37 people (23 male and 14 female) uttered 10 different surnames each, that is, we had 370 surnames to be recognized in total. We experimented with different values of N, both with and without phonological rules. The results are depicted in Table 12. In each cell the first value shows the absolute number of correct recognitions and the second the corresponding percentage.

If we do not use phonological rules, the best results are given when the recognizer produces the 30-best hypotheses. However, in this case the response time is quite increased, which necessitates a lower value of N. We have not kept record of the response time in all

Table 12. Surname recognition accuracy for different values of N (in the N-best hypotheses' list), with and without the application of phonological rules.

	Male (%)*	Female (%)	Total (%)
Without phonological rules			
$N = 1$	159/69.13	98/70.00	257/69.46
$N = 3$	162/70.43	98/70.00	260/70.27
$N = 5$	163/70.87	100/71.43	263/71.08
$N = 10$	168/73.04	102/72.85	270/72.97
$N = 15$	172/74.78	104/74.28	276/74.59
$N = 20$	179/77.82	108/77.14	287/77.56
$N = 25$	186/80.86	112/80.00	298/80.54
$N = 30$	191/83.04	116/82.85	307/82.97
With phonological rules			
$N = 1$	195/84.78	119/85.00	314/84.86
$N = 3$	200/86.95	121/86.43	321/86.75
$N = 5$	202/87.82	123/87.85	325/87.83
$N = 10$	207/90.00	127/90.71	334/90.27

*Percentages represent the second value of each column only.

these tests. Nevertheless, it was obvious that the system stopped being real-time with N greater than 3 because the computational cost became too high. When we applied phonological rules, we realized that $N = 1$ was enough to produce better results than $N = 30$ (without phonological rules), with no significant computational cost. This was due to the fact that the cost of processing the signal in order to produce multiple outputs is much higher than the time required for taking an output and applying the phonological rules. Moreover, the application of rules leads to significantly more than 30 solutions, which have the advantage of being based on language dependent data (not just the acoustic signal). Thus, the probability of including the correct surname is higher. The results are even better when we have $N = 10$ and use phonological rules. However, in this case, as for $N = 10$ without rules, the response time is not very good. In conclusion, $N = 3$ with phonological rules is the solution that combines good recognition accuracy and real-time response. In total, there were 52 rules, which is a high number if we consider that the structure of the rules allows for including many cases in the same rule by using classes. At first, we had 95 rules, but the processing time was prohibitive for real-time applications with no gain in accuracy because most of the rules covered very rare cases. Thus, we decided to keep only the ones that covered the most frequent interchanges between phonemes and phoneme combinations.

In this final improved version of IDAS, the HTK decoder has been replaced by the Philips SpeechPearl 2000 recognition engine. This is because now we are not only interested in research matters but also in having a commercial application for automating DAS as well as a platform for developing other speech-based human-interaction systems. Another difference compared to the D1.2 prototype is that the system prompts the user to give only the first letter of the surname. In order to decrease the average dialogue duration and number of turns that had quite high values in both D1.1 and D1.2, the confirmation dialogue states are discarded and empirically set confidence levels are applied. We use two confidence levels in the speech recognition process, *LEVEL_HIGH* and *LEVEL_LOW*. If the confidence level provided by the recognizer is greater than *LEVEL_HIGH*, the dialogue will continue with no problems. Otherwise the system will ask the user to repeat his/her request. If the recognition result of the second time is the same as the one produced the first time, the comparison level will be *LEVEL_LOW*. On the contrary, different recognition results for the first and second times entail that the new comparison level will be set to *LEVEL_HIGH*. Now if the recognition confidence level is greater than the set comparison level the dialogue will proceed, otherwise the control will be transferred to the human operator.

In Fig. 3, we can see the GUI module where the recognition result together with its confidence level are given for the current dialogue state and each one of the active channels. Each Dialogic card supports up to 4 channels. In Fig. 3, the first channel is in use and the second one ready for incoming calls. Channels 3 and 4 are not activated. In order to decide on the confidence levels the system is set in function using various confidence values and then the recognition results are compared with the real user utterances (that have been recorded and transcribed). The new confidence level will be set at the point where the mean verification error is minimum. The mean verification error is defined as the mean value of the false rejection and false acceptance errors. The values of *LEVEL_HIGH* and *LEVEL_LOW* are influenced by varying parameters, e.g., the quality of the telephone lines.

6. Summary and Conclusions

In this paper, we described a spoken dialogue system for automating DAS. Two prototypes D1.1 (basic approach) and D1.2 (improved version) were developed successively. In addition the system was extended so that it can be used under real-world conditions. The system architecture was presented in detail. Moreover evaluation experiments were carried out for all prototypes and the corresponding results were provided. In order to cope with the large vocabulary recognition issues, we used DAWGs and context-dependent phonological rules, which resulted in improved performance in terms of both response time and accuracy.

Currently the rules are formed manually, so our future work focuses on developing an algorithm for their automatic extraction that will exploit both linguistic and acoustic knowledge. In this way, we expect that we will cover cases not captured by the human designer using rules that are recognizer-dependent, while at the same time completely automating the process. Further experiments will be carried out concerning the optimization of the trade-off between recognition accuracy and response time. We will also experiment with different language models so that our system can handle successfully both grammatically correct utterances and spontaneous ungrammatical speech. Another issue that must be further investigated is the estimation of confidence levels. Finally, we plan to carry out field tests like the ones of the D1.2 prototype so that the overall performance of the new extended system is assessed.

Note

1. The system was developed in the framework of the EU project LE4-8315 IDAS (Interactive telephone-based Directory Assistance Services).

References

Aoe, J., Morimoto, K., and Hase, M. (1993). An algorithm for compressing common suffixes used in trie structures. *Systems and Computers in Japan*, *24*(12):31–42 (Translated from *Trans. IEICE*, *J75-D-II*(4):770–799, 1992).

Aust, H., Oerder, M., Seide, F., and Steinbiss, V. (1995). The Philips automatic train timetable information system. *Speech Communication*, *17*:249–262.

Betz, M. and Hild, H. (1995). Language models for a spelled letter recognizer. *Proceedings of ICASSP*, Detroit, MI, vol. 1, pp. 856–859.

Collingham, R.J., Johnson, K., Nettleton, D.J., Dempster, G., and Garigliano, R. (1997). The Durham telephone enquiry system. *International Journal of Speech Technology*, *2*(2):113–119.

Córdoba, R., San-Segundo, R., Montero, J.M., Colás, J., Ferreiros, J., Macías-Guarasa, J., and Pardo, J.M. (2001). An interactive directory assistance service for Spanish with large-vocabulary recognition. *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 1279–1282.

Daciuk, J., Mihov, S., Watson, B., and Watson, R. (2000). Incremental construction of minimal acyclic finite state automata. *Computational Linguistics*, *26*(1):3–16.

Gao, Y., Ramabhadran, B., Chen, J., Erdoğan, H., and Picheny, M. (2001). Innovative approaches for large vocabulary name recognition. *Proceedings of ICASSP*, Salt Lake City, Utah.

Gardner-Bonneau, D. (1992). Human factors problems in interactive voice response (IVR) applications: Do we need a guideline/standard? *Proceedings of Human Factors Society, 36th Annual Meeting*, vol. 1, pp. 222–226.

Georgila, K., Tsopanoglou, A., Fakotakis, N., and Kokkinakis, G. (1998). An integrated dialogue system for the automation of call centre services. *Proceedings of ICSLP*, Sidney, Australia, pp. 45–48.

Georgila, K., Sgarbas, K., Fakotakis, N., and Kokkinakis, G. (2000). Fast very large vocabulary recognition based on compact DAWG-structured language models. *Proceedings of ICSLP*, Beijing, China, vol. 2, pp. 987–990.

Georgila, K., Fakotakis, N., and Kokkinakis, G. (2001a). Efficient stochastic finite-state networks for language modelling in spoken dialogue systems. *Proceedings of Eurospeech*, Aalborg, Denmark, vol. 1, pp. 247–250.

Georgila, K., Tsopanoglou, A., Fakotakis, N., and Kokkinakis, G. (2001b). Improved large vocabulary speech recognition using lexical rules. *Proceedings of PCHCI—Advances in Human-Computer Interaction*, Patras, Greece, pp. 191–196.

Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., and Zue, V. (1995). Multilingual spoken-language understanding in the MIT Voyager system. *Speech Communication*, *17*:1–18.

Gong, L. and Lai, J. (2001). Shall we mix synthetic speech and human speech? Impact on users' performance, perception and attitude. *Proceedings of CHI*, pp. 158–165.

Gorin, A., Riccardi, G., and Wright, J.H. (1997). How May I Help You? *Speech Communication*, *23*:113–127.

Gupta, V., Robillard, S., and Pelletier, C. (1998). Automation of locality recognition in ADAS plus. *Proceedings of IVTTA*, Turin, Italy, pp. 1–4.

Hanazawa, K., Minami, Y., and Furui, S. (1997). An efficient search method for large-vocabulary continuous-speech recognition. *Proceedings of ICASSP*, Munich, Germany, pp. 1787–1790.

Hennecke, M.E., Kaspar, B., Tsopanoglou, A., Michos, S., Mantakas, M., and Safra, S. (1999). Design specification and planning of evaluation (IDAS Technical Report 2.2:D1.2).

Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Fosler, E., and Morgan, N. (1994). The Berkeley restaurant project. *Proceedings of ICSLP*, pp. 2139–2142.

Kamm, C.A., Shamieh, C.R., and Singhal, S. (1995). Speech recognition issues for directory assistance applications. *Speech Communication*, *17*:303–311.

Kaspar, B. et al. (1997). SPRADIAK—Directory assistance pilot. *Proceedings of VOICE.*

Lamel, L., Rosset, S., Gauvain, J.L., Bennacef, S., Garnier-Rizet, M., and Prouts, B. (2000). The LIMSI ARISE system. *Speech Communication*, *31*:339–353.

Lennig, M. (1990). Putting speech recognition to work in the telephone network. *IEEE Computer*, *23*(8):35–41.

Lennig, M., Bielby, G., and Massicotte, J. (1995). Directory assistance automation in Bell Canada: Trial results. *Speech Communication*, *17*:227–234.

Rahim, M., Di Fabbrizio, G., Kamm, C., Walker, M., Pokrovsky, A., Ruscitty, P., Levin, E., Lee, S., Syrdal, A., and Schlosser, K. (2001). Voice-IF: A mixed-initiative spoken dialogue system for AT&T conference services. *Proceedings of Eurospeech*, Aalborg, Denmark, vol. 2, pp. 1339–1342.

Ramabhadran, B., Bahl, L.R., de Souza, P.V., and Padmanabhan, M. (1998). Acoustics-only based automatic phonetic baseform generation. *Proceedings of ICASSP*, Seatlle, WA, vol. 1, pp. 309–312.

Schmid, P., Cole, R., and Fanty, M. (1993). Automatically generated word pronunciations from phoneme classifier output. *Proceedings of ICASSP*, Minneapolis, MN, vol. 2, pp. 223–226.

Seide, F. and Kellner, A. (1997). Towards an automated directory information system. *Proceedings of Eurospeech*, Rhodes, Greece, vol. 3, pp. 1327–1330.

Sgarbas, K., Fakotakis, N., and Kokkinakis, G. (1995). Two algorithms for incremental construction of directed acyclic word graphs. *International Journal on Artificial Intelligence Tools*, *4*(3):369–381.

Sgarbas, K., Fakotakis, N., and Kokkinakis, G. (2001). Incremental construction of compact acyclic NFAs. *Proceedings of ACL-EACL*, Toulouse, France, pp. 482–489.

Sugamura, N., Hirokawa, T., Sagayama, S., and Furui, S. (1998). Speech processing technologies and telecommunications applications at NTT. *Proceedings of IVTTA*, Turin, Italy, pp. 37–42.

Van den Heuvel, H., Moreno, A., Omologo, M., Richard, G., and Sanders, E. (2001). Annotation in the SpeechDat projects. *International Journal of Speech Technology*, *4*(2):127–143.

Whittaker, S.J. and Attwater, D.J. (1995). Advanced speech applications—The integration of speech technology into complex services. *ESCA Workshop on Spoken Dialogue Systems—Theory and Application*, Visgø, Denmark, pp. 113–116.

Young, S., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1997). *The HTK Book*, user manual, Entropic Cambridge Research Laboratory, Cambridge.

Zue, V., Seneff, S., Glass, J., Hetherington, L., Hurley, E., Meng, H., Pao, C., Polifroni, J., Schloming, R., and Schmid, P. (1997). From interface to content: Translingual access and delivery of on-line information. *Proceedings of Eurospeech*, Rhodes, Greece, vol. 4, pp. 2227–2230.