Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Speech and Language 20 (2006) 420–440
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

Combining language models in the input interface of a spoken dialogue system

R. López-Cózar *, Z. Callejas

Department Languages and Computer Systems, Computer Science Faculty, Granada University, 18071 Granada, Spain

Received 4 February 2004; received in revised form 9 December 2004; accepted 24 May 2005
Available online 22 June 2005

Abstract

This paper presents a new technique to enhance the performance of the input interface of spoken dialogue systems based on a procedure that combines during speech recognition the advantages of using prompt-dependent language models with those of using a language model independent of the prompts generated by the dialogue system. The technique proposes to create a new speech recognizer, termed *contextual speech recognizer*, that uses a prompt-independent language model to allow recognizing any kind of sentence permitted in the application domain, and at the same time, uses contextual information (in the form of prompt-dependent language models) to take into account that some sentences are more likely to be uttered than others at a particular moment of the dialogue. The experiments show the technique allows enhancing clearly the performance of the input interface of a previously developed dialogue system based exclusively on prompt-dependent language models. But most important, in comparison with a standard speech recognizer that uses just one prompt-independent language model without contextual information, the proposed recognizer allows increasing the word accuracy and sentence understanding rates by 4.09% and 4.19% absolute, respectively. These scores are slightly better than those obtained using linear interpolation of the prompt-independent and prompt-dependent language models used in the experiments.

* Corresponding author. Tel.: +34 958 240579/243271; fax: +34 958 243179/243230.
E-mail addresses: rlopezc@ugr.es (R. López-Cózar), zoraida@correo.ugr.es (Z. Callejas).

0885-2308/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2005.05.003

1. Introduction

Spoken dialogue systems are computer programs developed to interact with users using speech in order to provide specific services automatically, mainly through the telephone line, as for example travel information (Seneff and Polifroni, 2000; Lamel et al., 2000), language learning (Ehsani et al., 2000), car-driver assistance (Bernsen, 2003; Baca et al., 2003), weather information (Zue et al., 2000; Wang et al., 2000; Nakano et al., 2001), automatic call-routing (Huang and Cox, 2003; Fegyó et al., 2003), etc. Several systems have also been set up for the fast food domain, as for example TOSBURG II (Seto et al., 1994) and SAPLEN, the later developed in our laboratory (López-Cózar et al., 2000, 2001, 2002, 2003). Fig. 1 shows the structure of the SAPLEN system, typical of current spoken dialogue systems (Pellom et al., 2000; Filisko and Seneff, 2003), consisting of input interface (speech recognizer and semantic analyzer), dialogue manager, memory module, database and output interface (response generator and synthesizer).

The speech recognizer was created using the Hidden Markov Model Toolkit (HTK) (Hain et al., 1999; Young et al., 2000). This system module converts the user voice (utterance) into a sequence of words included in the system dictionary, considering a language model (concretely a word bigram) that determines all the possible sentences that can be recognized. The recognizer outputs (recognized sentences) are converted by the semantic analyzer into frame representations that capture the meaning of the sentences (Niimi et al., 2000; Bonneau-Maynard and Rosset, 2003). These representations get stored in the system memory module. The analyzer uses 45 semantic rules based on the detection of keywords and on certain expressions in sentences (Kawahara et al., 1998; Zhang et al., 1998). It uses an implicit recovery strategy that sometimes allows the correct semantic interpretation to be obtained even if some words in the recognizer output are wrongly recognized. This strategy allows to recover from meaningless-word recognition errors, since these words do not change the semantic content of sentences. To implement this strategy the analyzer discards, on the one hand, meaningless words (disfluencies, articles and prepositions) when processing the

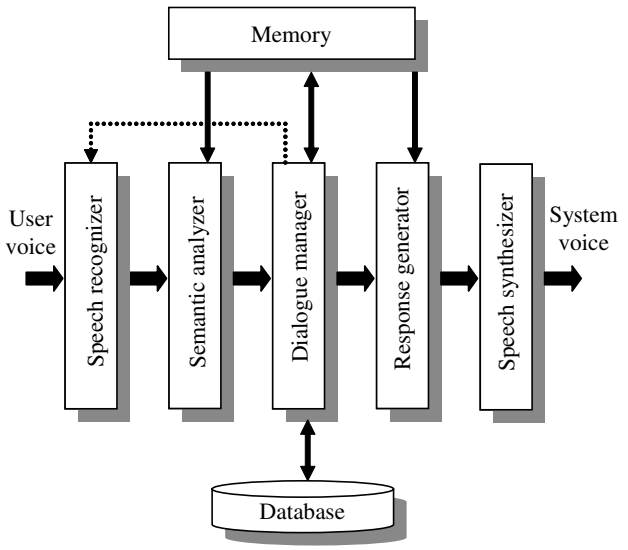

Fig. 1. Structure of the SAPLEN dialogue system.

sentences. Thus, the sample sentence "please ... uhm ... two big colas ... but without ice" is analyzed considering only the keywords "two big colas without ice", which allow obtaining the correct meaning. On the other hand, the analyzer discards Spanish gender/number discordances caused due to the southern Spain accent of most users. Due to this accent, the sentence "two beers" might be recognized as "two beer" since the final 's' of plural words usually is not pronounced. To implicitly recover this kind of error, the analyzer focuses on the number ("two" in this case) thus obtaining the correct semantic representation (number = 2, product = beer).

The dialogue manager is the core of the system. It decides the next action to be taken (Heeman et al., 2003), the next prompt to be generated and the expectations for the speech recognizer about the sentence type the user will likely utter in his next turn (dotted arrow in Fig. 1). The dialogue strategy is as follows: the system initially prompts the user to order products and when an order is made it prompts for his telephone number. The system accesses a database to find out whether the user is already known. If so, it prompts to confirm the data stored in the database, and otherwise it asks for the post code and the address, and stores the new user data in the database. At the end of the dialogue, the system prompts to confirm the products ordered, the price and the estimated delivery time. After obtaining each data item the system generates an explicit confirmation if it has low confidence on the obtained data (e.g., "Did you say your telephone number is 9 5 8 1 2 3 4 5 6?"), and otherwise it includes an implicit confirmation in the prompt to obtain the next data item (e.g., "Ok, telephone number 9 5 8 1 2 3 4 5 6. What is your post code?") (Wang and Lin, 2003). The user is told at the beginning of the dialogue that to correct possible system errors he must utter expressions such as "error", "it is wrong" or "you made a mistake", which are permitted (i.e., recognizable) in every state. These utterances make the system go back to the previous dialogue state, so that it will prompt again for the data item wrongly obtained. The confidence score of a keyword w is obtained from the N-best output ($N = 10$) of the speech recognizer as follows:

$$C(w) = \frac{\sum_{i \in I_w} \exp(-sc_i)}{\sum_{j=1}^{N} \exp(-sc_j)}, \tag{1}$$

where sc_j represents the score (log probability) of the jth hypothesis in the N-best list, I_w represents the set of indices of the hypotheses containing the keyword ($I_w \subset \{1, \ldots, N\}$), and sc_i represents the score of the ith hypothesis containing the keyword. The confidence score of data items containing just a keyword, e.g., size (small, large, etc.) or taste (orange, lemon, etc.) is the confidence score of the keyword, while the score of data items containing several keywords (e.g., a telephone number, post code, etc.) is the lowest score of the keywords (digits in these cases).

Finally, the output interface of the system contains the response generator which builds the system response in text format, and the synthesizer that transforms it into the system's voice (Takeuchi et al., 2003).

1.1. Prompt-dependent and prompt-independent language models

There are dialogue systems designed to recognize sentences by considering a specific language model associated with each system prompt decided by the dialogue manager (Lane et al., 2003; Mori et al., 2003), which is the case, e.g., of SAPLEN. These language models, called *prompt-*

dependent language models in this paper, aim to provide high speech recognition rates and are useful if the interaction is clearly constrained by the system; however, they are not adequate if the user does not follow the system indications and utters sentences not permitted by these language models. For example, if the system generates the prompt "What is your telephone number?" and the user actually utters a telephone number then the sentence can be correctly recognized, but if he utters a different kind of sentence (an address, for instance) then the recognizer output will be any of the possible telephone numbers and the address will never be recognized. Consequently, the user may feel uncomfortable during the interaction as he perceives that any deviation from the system indications provokes a system malfunction. As an attempt to solve this problem, other dialogue systems use a general language model that is used during the whole dialogue, instead of using a particular language model associated with every prompt. This language model is prompt independent and is designed to recognize any kind of sentence within the application domain, which aims to provide users with a more comfortable and natural interaction. However, this language model tends to have higher perplexity and its vocabulary is generally significantly larger, which may lead to increase recognition errors and provoke a system malfunction.

The rest of the paper is organized as follows. Section 2 presents the technique proposed in this paper. It includes references to previous related work, defines word-networks and word-class bigrams, and describes how these bigrams can be mapped to the prompts of a dialogue system. The section concludes showing the procedure used to analyze the word-networks in order to provide recognized sentences. The experimental results are presented in Section 3. It initially sets out a description of the test and training corpora, word-networks and word-class bigrams used. Then it addresses the performance of the initial input interface of the dialogue system for the in-context and out-of-context sentence analysis, and then focuses on the use of the technique proposed in this paper for both types of analysis. The section also reports on experiments carried out using linear interpolation instead of the proposed technique to re-score the word-networks, and ends by noting some limitations of the technique. Finally, the conclusions and some lines for future work are presented in Section 4.

2. The proposed technique: contextual speech recognizer

The technique presented in this paper proposes to create the so-called *contextual speech recognizer* shown in Fig. 2, which is a compound of two modules. On the one hand, a standard speech recognizer that receives the user voice (utterance) and produces a word-network using acoustic models previously trained from a speech database, and a language model (word bigram[1] in our setting) previously compiled from a sentence corpus. On the other hand, the proposed recognizer contains a WN-analyzer (word-network analyzer) that receives the word-network and produces the recognized sentence considering the current prompt type of the dialogue system (T_i), a "probability increment" parameter (p) that increments the probabilities of determined transitions in the word-network, and a word-class bigram mapped to the prompt type, as will be explained in Section 2.4.

[1] A word bigram is a particular case of n-gram where $n = 2$. It is used to estimate the probability $P(w_i|w_{i-1})$, i.e., the probability of the word in position i in a sentence given the word in position $i - 1$.

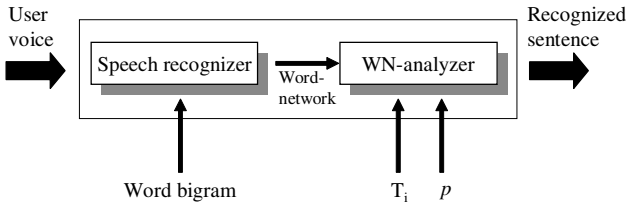

Fig. 2. Contextual speech recognizer.

The technique combines the advantages of using a general language model (used to create the word-network) and a prompt-dependent language model (mapped to the prompt type T_i) in speech recognition, since the word-network is analyzed taking into account that certain words and sentence types are more likely to be uttered at a given moment of the dialogue. Then it reduces the word error rate by restricting the words and expressions considered, and helps provide a more comfortable interaction for users since they can utter any kind of sentence to answer prompts without provoking a system malfunction. In this paper, the sentences pronounced by the user according to the system prompt are said to be *in-context* analyzed (for example, the system prompts for a telephone number and the user actually utters a telephone number). If the user answers the prompt with another type of sentence (an address, for instance) then the sentence is said to be *out-of-context* analyzed.

2.1. Previous related work

Several papers in the literature have addressed the enhancement of speech recognition systems using an additional module that processes the recognizer output with additional information to reduce recognition errors. For example, (Gaudinant et al., 1999) present a recognition system that combines standard HMM (Hidden Markov Models) recognition with a linguistic parser. The additional module processes a network of phonemes produced by a front-end HMM recognizer and produces the best word sequence according to linguistic information. This module is a compound of a lexical analyzer, which reads the phoneme sequence and produces a network of lexical items, and a syntactic parser that builds syntactic structures on the basis of the word hypothesis, filtering out ungrammatical combinations of words. The technique we propose also uses an additional module that processes the output of a HMM recognizer and produces a recognized sentence. However, in our approach the recognizer output is a network of words instead of a network of phonemes and the ungrammatical combinations of words are not filtered out at the moment.

The improvement of speech recognition systems has also been addressed by combining word *n*-grams and word-class *n*-grams (Siu and Ostendorf, 2000; Niesler and Woodland, 1999). Word *n*-grams generally provide high performance if a large amount of training data is available; however, the performance decreases if the training data are insufficient. On the contrary, word-class *n*-grams provide better results when less training data are available. An interpolation method to integrate both types of *n*-grams can be found in Kobayashi and Kobayashi, 1999. The technique we present in this paper also uses both types of *n*-grams, i.e., word bigrams and word-class bigrams. In our approach, a word bigram compiled using a sentence

corpus, is used by a speech recognizer to process every input utterance and generate a word-network. Later, the WN-analyzer takes the word-network, uses a word-class bigram associated with the current prompt type of the dialogue system, analyzes the network considering the bigram and finally produces a recognized sentence.

Another way to enhance speech recognition is based on information reliant on the dialogue state. For example, Visweswariah and Prints (2001) observe that when a user converses with a dialogue system, the state of the dialogue strongly influences the responses expected from the user; moreover, the prompts generated by the system can play an important role with respect to the language model. The authors report that by using information about the dialogue state, the word error rate can be reduced by about 9%. The technique we propose is also concerned with using the dialogue state to enhance speech recognition and understanding in spoken dialogue systems, since the word-class bigram the WN-analyzer uses to process the word-network is associated with a prompt type of the dialogue system, and this prompt type is associated with the dialogue state.

2.2. Word-networks

A word-network consists of a list of nodes and a list of arcs, the nodes represent words and the arcs represent transitions between words. Fig. 3 shows an excerpt of a word-network used in the experiments, which allows recognizing sentences in the fast food domain (e.g., "one small beer", "one red wine", etc.). The network arcs have a language probability (l) and an acoustic probability (a) assigned.

The HTK provides a standard format for representing word-networks, called SLF (Standard Lattice Format). Using this format, the network shown in Fig. 3 can be represented as shown in Fig. 4, in which "N" represents the number of network nodes, "L" the number of network arcs (transitions), "I" the node number, "W" the word in the node, "J" the arc number, "S" the transition start node, "E" the transition end node, and finally "a" and "l" the transition acoustic and language log probabilities, respectively (the figure omits some data unnecessary for the explanation).

2.3. Word-class bigrams

The proposed technique uses word-class bigrams (C_i's) to exploit grammatical information concerned with syntactic patterns (Emami, 2003). An important reason for using these bigrams instead of word bigrams is that they allow generalizing word-pair sequences not seen in the sentence corpus used for training (Zitouni et al., 2003). The word-class bigrams can be obtained in three steps using a corpus of dialogues, either human-to-human, human-to-system or simulated using the Wizard of Oz technique (Dahlbäck et al., 1993). The dialogues must be concerned with the application domain for which the dialogue system is being (or has been) designed. The goal of the first step is to classify the client (or user) sentences into sentence types U_i uttered to answer questions (or prompts) T_i generated by the human operator (or the dialogue system). For example, in the fast food domain we could consider sentence types such as product orders, telephone numbers, post codes, addresses, etc. The goal of the second step is to find out the keywords in these sentences and classify them into word-classes W_k considering the type of word (e.g., Table 1 shows some possible word-classes in the fast food domain).

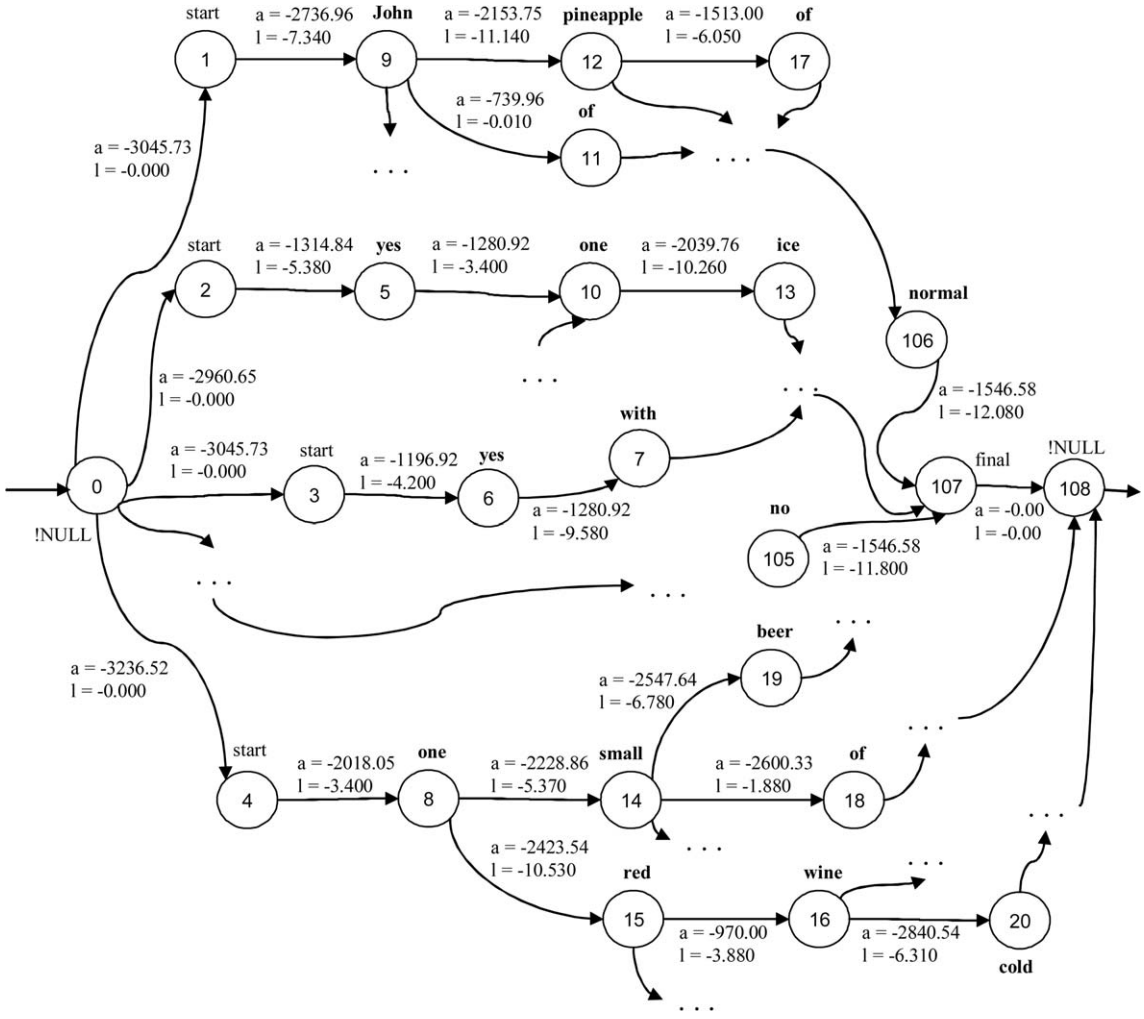

Fig. 3. Excerpt of a word-network.

The third step can be carried out using an automatic procedure that takes each set of sentences U_i created in the first step and substitutes the keywords for the word-classes the keywords belong to. This step transform each sentence into a sequence of word-classes, possibly including meaningless words among them. After the transformation, the procedure analyzes each obtained sequence and adds each adjacent pair of word-classes $(W_k,\ W_{k+1})_{k=1,\ldots,n-1}$ to a set S_i associated with the sentences U_i, where n is the number of word-classes in the obtained sequence and W_k is the word-class in position k in the sequence. Finally, a word-class bigram C_i is created from S_i. For example, suppose U_l is a set of food orders uttered to answer the prompt "What would you like to have?" and assume it contains the sentences "uhm two ham sandwiches" and "two beers please". Taking into account the sample word-classes shown in Table 1 and applying this procedure, the sentences are transformed into "uhm NUMBER INGREDIENT

```
N=109  L=338
I=0    W=!NULL
I=1    W=start
I=2    W=start
I=3    W=start
I=4    W=start
I=5    W=yes
I=6    W=yes
I=7    W=with
I=8    W=one
I=9    W=John
I=10   W=one
I=11   W=of
I=12   W=pineapple
I=13   W=ice
I=14   W=small
I=15   W=red
I=16   W=wine
I=17   W=of
I=18   W=of
I=19   W=beer
I=20   W=cold
…
I=105  W=no
I=106  W=normal
I=107  W=final
I=108  W=!NULL
J=0    S=0    E=1    a=-3045.73    l=-0.000
J=1    S=0    E=2    a=-2960.65    l=-0.000
J=2    S=0    E=3    a=-3045.73    l=-0.000
J=3    S=0    E=4    a=-3236.52    l=-0.000
J=4    S=1    E=9    a=-2736.96    l=-7.340
J=5    S=9    E=12   a=-2153.75    l=-11.140
J=6    S=12   E=17   a=-1513.00    l=-6.050
J=7    S=9    E=11   a=-739.96     l=-0.010
J=8    S=2    E=5    a=-1314.84    l=-5.380
J=9    S=5    E=10   a=-1280.92    l=-3.400
J=10   S=10   E=13   a=-2039.76    l=-10.260
J=11   S=3    E=6    a=-1196.92    l=-4.200
J=12   S=6    E=7    a=-1280.92    l=-9.580
J=13   S=4    E=8    a=-2018.05    l=-3.400
J=14   S=8    E=14   a=-2228.86    l=-5.370
J=15   S=14   E=18   a=-2600.33    l=-1.880
J=16   S=14   E=19   a=-2547.64    l=-6.780
J=17   S=8    E=15   a=-2423.54    l=-10.530
J=18   S=15   E=16   a=-970.00     l=-3.880
J=19   S=16   E=20   a=-2840.54    l=-6.310
…
J=335  S=106  E=107  a=-1546.58    l=-12.080
J=336  S=105  E=107  a=-1546.58    l=-11.800
J=337  S=107  E=108  a=-0.00       l=-0.00
```

Fig. 4. Excerpt of a word-network represented in SLF format.

Table 1
Examples of word-classes in the fast food domain

Word-class	Word examples
NUMBER	one, two, three, four, five, six, ...
FOOD	sandwich, cake, ice-cream, hamburger, sandwiches, cakes, ice-creams, hamburgers, ...
INGREDIENT	cheese, ham, bacon, apple, ...
DRINK	water, beer, coke, wine, beers, cokes, wines, ...
SIZE	small, large, big, ...
TASTE	orange, lemon, apple, ...

FOOD'' and ''NUMBER DRINK please''.[2] So that, the automatic procedure adds the pairs (NUMBER, INGREDIENT) and (INGREDIENT, FOOD) from the first sentence, and the pair (NUMBER, DRINK) from the second to S_l. If a particular word belongs to several word-classes the corresponding W_k refers to all them; for example, the word sequence ''one apple'' would be transformed into two word-class pairs: (NUMBER, INGREDIENT) and (NUMBER, TASTE), since the word ''apple'' belongs to the word-classes INGREDIENT and TASTE. The word-class bigram C_l is created from the set S_l.

After the word-class bigrams (C_i's) have been built from the analysis of a dialogue corpus, the technique presented in this paper proposes to create a set Ω of mappings between the prompt types T_i the dialogue system generates and the word-class bigrams C_i previously created:

$$\Omega = \{T_i, C_i\}_{i=1,\ldots,m},$$

where m represents the number of prompt types and C_i represents the word-class bigram associated with T_i. For example, if the dialogue system only generates prompt types to enter product orders (e.g., ''What would you like to order?'', ''Please say what you want to have''), telephone numbers (e.g., ''Please say your telephone number'', ''Please say your telephone number again''), post codes (e.g., ''What is you post code?'', ''Please repeat your post code'') and addresses (e.g., ''What is your address?'', ''Please say your address again''), then $m = 4$. For these prompt types we should create the sets $U_{i=1,\ldots,4}$ and $S_{i=1,\ldots,4}$, and then compile the word-class bigrams $C_{i=1,\ldots,4}$.

2.4. Procedure to analyze word-networks

To process each word-network the WN-analyzer uses the word-class bigram C_i mapped to the prompt type T_i and the ''probability increment'' parameter (p), increasing the probability of the transition $w_S \rightarrow w_E$ if there is a word-class pair (W_K, W_L) in C_i, with w_S in W_K, w_E in W_L. For example, the probability of the transition ''one → ham'' is increased if (NUMBER, INGREDIENT) is in C_i. The procedure to analyze word-networks is described algorithmically in Fig. 5.

Initially the procedure selects the word-class bigrams C_i mapped to the prompt type T_i and stores in ''Transitions'' the set of all the transitions $w_S \rightarrow w_E$ in the word-network. Secondly it adds two additional fields to each node in the word-network: ''prob'' and ''previous_word''.

[2] In the first transformed sentence $n = 3$, W_1 = NUMBER, W_2 = INGREDIENT and W_3 = FOOD, while in the second $n = 2$, W_1 = NUMBER and W_2 = DRINK.

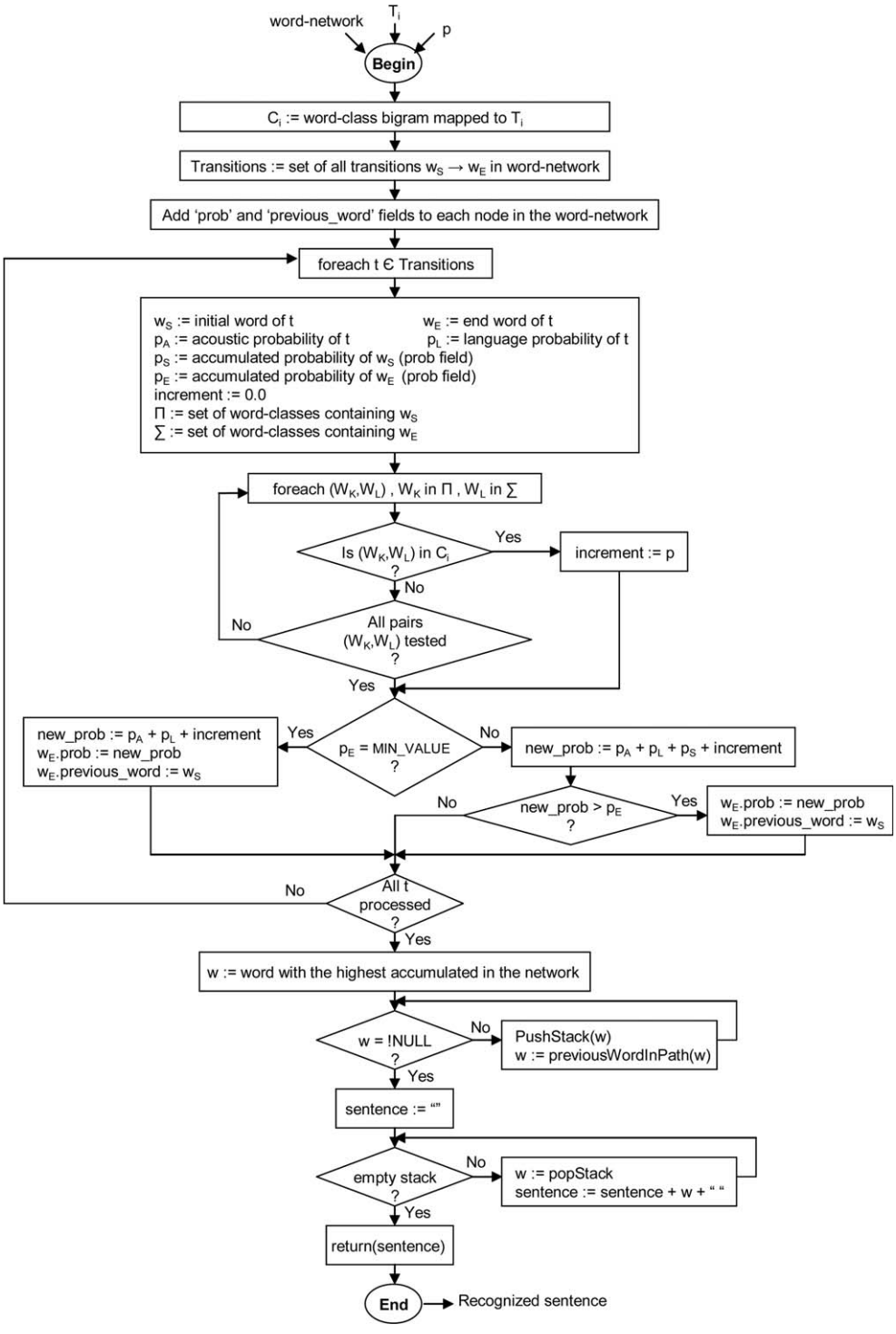

Fig. 5. Procedure to analyze word-networks.

The "prob" field represents the accumulated probability of the word in the best path to be created, and is initialized with the special value MIN_VALUE which represents the word has not assigned a probability yet. The "previous_word" field stores a pointer to the previous word in the best path, and is initialized with the NULL value. The main loop of the procedure creates the best path in the word-network by analyzing all the transitions. It initially makes some definitions depending on the current t transition being analyzed and then checks whether there is a word-class pair (W_K, W_L) in the word-class bigram C_i mapped to the prompt T_i, with w_S in W_K, w_E in W_L. If yes, the "increment" variable is assigned the value of the p parameter. Next, the loop checks whether the transition end word (w_E) has already been visited. If not (case p_E = MIN_VALUE) it is assigned the sum of the transition acoustic and language probabilities, and the increment value (either 0.0 or p), while the pointer to the "previous word" is set to point to the w_S word. If w_E was visited previously, the procedure checks whether the sum of the transition acoustic and language probabilities, the increment value and the current accumulated probability of the w_S word is greater that the current accumulated probability of the w_E word. If yes, the accumulated probability and the pointer to the "previous word" are updated (as mentioned before) and otherwise no change is made. Once this processing has been made for all the transitions in the word-network, the procedure locates the word with the greatest accumulated probability and follows back the "previous word" pointer sequence until reaching a node that contains the special word !NULL (initial node of the word-network). This way, words are visited in reverse order (from the end to the start); thus, they are temporarily stored in a stack structure to be outputted in the correct order as the recognized sentence.

As can be seen in Fig. 2, the word-network is processed by the NW-analyzer, which uses the procedure described above. We remind that the word-network is produced by a speech recognizer that uses a general language model (in our setting a *word bigram*) compiled from a sentence corpus that contains all types of sentence permitted (e.g., product orders, telephone numbers, etc.). Hence, using this procedure any permitted sentence can be recognized (in theory) independently of the current prompt of the dialogue system, which enables out-of-context sentences to be correctly recognized. The use of a *word-class bigram* in the WN-analyzer is the prompt-dependent part of the proposed technique. This bigram is used to favor the sentence recognition (and understanding) of determined sentences, taking into account the context of the dialogue. For example, if the system generates the prompt "What is your telephone number?", the procedure favors (increasing probabilities) the recognition of sentences such as "9 5 8 1 2 3 4 5 6" since the word-class bigram used was compiled precisely from this type of sentences, whereas it allows recognizing any other sentence type permitted in the domain (e.g., "I want a ham sandwich") since the word bigram used was compiled from all the sentence types in the domain.

3. Experiments

The goal of the experiments is triple. Firstly, to compare the performance of the initial input interface of the SAPLEN system with that of a new input interface in which the proposed contextual speech recognizer is used. Secondly, to show the advantage of using the proposed recognizer instead of a standard speech recognizer that uses one general prompt-independent language model but does not use at the same time prompt-dependent contextual information,

i.e., prompt-dependent language models. Thirdly, to compare the performance of the proposed procedure to re-score the word-networks (probability increment) with that of using linear interpolation of the prompt-independent word bigram and the prompt-dependent word-class bigrams used in these experiments. Evaluation metrics are word accuracy (WA) and speech understanding (SU) rates. The WA is calculated as $WA = (w_t - w_i - w_s - w_d)/w_t$, where w_t is the total number of words in the sentences, and w_i, w_s and w_d are the number of words inserted, substituted and deleted by the recognizer, respectively. The SU is calculated as $SU = S_u/S_t$, where S_u is the number of sentences correctly analyzed semantically by the semantic analyzer and S_t is the total number of sentences.

3.1. Description of the corpora used in the experiments

In order to develop the SAPLEN system we previously collected a dialogue corpus in a fast food restaurant that contains about 800 recorded dialogues in Spanish regarding telephone conversations between clients and restaurant assistants (López-Cózar et al., 1998). These dialogues contain product orders, telephone numbers, post codes, addresses, queries, confirmations, greetings and other types of sentence. The dialogues were transcribed and analyzed for previous works (López-Cózar et al., 2000, 2001, 2002, 2003) including tags regarding the speakers, sentence types, pragmatic function of sentences and other kind of information (Hardy et al., 2003). Selecting at random 5250 client sentences among the 17 sentence types shown in Table 2, we have created two separate sentence corpora, one for training and the other for testing (with no training sentences observed in the test corpus).

The training corpus contains 4250 sentences, 250 sentences of each type shown in Table 2. We have also included in this corpus the 4250 sentence orthographic transcriptions as well as their corresponding 4250 semantic representations. The word-classes in these sentences were obtained

Table 2
Sentence types used in the experiments

Type	Sentence description	Sentence number
U_1	Product order	500
U_2	Telephone number	500
U_3	Post code	500
U_4	Address	500
U_5	Query	250
U_6	Confirmation	250
U_7	Amount	250
U_8	Food name	250
U_9	Ingredient	250
U_{10}	Drink name	250
U_{11}	Size	250
U_{12}	Taste	250
U_{13}	Temperature	250
U_{14}	Street name	250
U_{15}	Building number	250
U_{16}	Apartment floor	250
U_{17}	Apartment letter	250

for previous works, analyzing manually the transcriptions (Table 3 sets out some of these word-classes, translated from Spanish to English). Using the transcriptions we have compiled a word bigram to be used by the contextual speech recognizer to obtain the recognized sentences.

The test corpus is used to evaluate both the initial and the new input interfaces of the system. It contains 1000 sentences, 250 of each of the first four sentence types shown in Table 2 (i.e., 250 product orders, 250 telephone numbers, 250 post codes and 250 addresses). We have focused only on these four sentence types in the testing to obtain experimental results for four realistic cases of out-of-context sentence analysis that, taking into account the dialogue strategy of the system, may occur when the users try to correct system errors. Considering the dialogue strategy explained in Section 1, it is possible the system makes a false positive error (i.e., a data item wrongly recognized obtains a high confidence score), the system generates an implicit confirmation, and the user tries to correct the error when the system is prompting for a different data item, but forgets about the "go-back" navigation commands ("error", "it is wrong", "you made a mistake") and rephrases the sentence instead. Thus the rephrased sentence is out-of-context analyzed, and gets always wrongly recognized and understood, causing confusion and frustration in the user. Table 4 shows the eight cases of out-of-context analysis considered. For instance, the first case occurs if the system misunderstands a food order and the user tries to correct the error rephrasing the order instead of uttering a go-back command when the system is prompting for the telephone number, which may be illustrated as follows:

Table 3
Examples of word-classes

Word-class	# different words in word-class	Example words
NUMBER	103	zero, one, two, ...
FOOD_NAME	6	sandwich, cake, ...
INGREDIENT	28	ham, cheese, ...
DRINK_NAME	16	beer, wine, ...
SIZE	6	small, large, ...
TASTE	6	orange, lemon,...
TEMPERATURE	6	cold, hot,...
ADDRESS_ TYPE	5	street, square,...
STREET_NAME	324	elm, melrose,...
BUILDING_FLOOR	20	first, second,...
APARTMENT_ LETTER	28	a, b, c, d, e,...

Table 4
Cases of out-of-context sentence analysis

Case	Prompt type T_i	Sentence type U_i
1	Telephone number	Product order
2	Confirmation	Product order
3	Post code	Telephone number
4	Confirmation	Telephone number
5	Address	Post code
6	Confirmation	Post code
7	Post code	Address
8	Confirmation	Address

User: One ham sandwich please.
System: Ok, one cheese sandwich. What is your telephone number?
User: I said I want a ham sandwich.
System: Did you say your telephone number is 5 6 5 1 4 1 6 8 6?
User: I didn't say any telephone number, I just said I want a ham sandwich.
System: Did you say your telephone number is 5 6 5 1 4 1 6 8 6? ...

The prompt type "Telephone number" in the Table refers to the system prompts "What is your telephone number?", "Please say your telephone number"; the prompt type "Confirmation" refers to prompts such as "Did you say your telephone number is 9 5 8 1 2 3 4 5 6?", "Did you say a ham sandwich?", etc.; the prompt type "Post code" refers to the prompts "What is your post code?", "Please say your post code"; and the prompt type "Address" refers to the prompts "What is your address?", "Please say your address". The sentence types U_i refer to the sentences used to answer these prompts.

3.2. Experimental results

3.2.1. Initial input interface for in-context and out-of-context analysis

We firstly have tested the initial input interface of the SAPLEN system, which does not use the WN-analyzer. This interface is comprised of a standard HTK-based recognizer and a semantic analyzer. The speech recognition was based on prompt-dependent language models (word bigrams) using the Katz's back-off smoothing technique to estimate the probability of word pairs unseen in the training corpus (Katz, 1987). As commented in the previous section, we have focused on the first four sentence types shown in Table 2 (product orders, telephone numbers, post codes and addresses) and have compiled a word bigram for each sentence type using one half of the sentences (i.e., a bigram has been compiled from 250 product orders, other from 250 telephone numbers, etc.) whereas the other half has been used for testing (test sentences have not been observed in the training sentences). To decide the type of analysis (either in-context or out-of-context) we set manually an internal parameter of the interface that decides the bigram to use during the sentence analysis. Table 5 sets out the average results obtained when the sentences in the test corpus (1000 in total) are analyzed in-context (sentence type = prompt type) and out-of-context (considering the eight cases shown in Table 4).

As can be observed, the performance of the interface is acceptable when it analyzes sentences in-context but is totally unacceptable for the out-of-context analysis since the sentences are not correctly recognized (and consequently are not understood by the system). Almost all WA scores are even negative due to the high rate of insertion recognition errors. The output of the recognizer is a sentence permitted by the current bigram; thus, if for example the grammar was compiled from telephone numbers, the output is a telephone number independently of the sentence type actually analyzed. Note that for the product orders the SU score is higher than the WA score, which indicates the implicit recovery strategy used by the semantic analyzer is more useful for this sentence type than for the others. As it was discussed in Section 1, this strategy allows recovering from meaningless words and gender/number discordances in Spanish sentences (e.g., "two sandwich"). Thus, the result indicates some product orders are correctly understood even though some words are wrongly recognized. The recovery strategy is not useful at all for telephone numbers

Table 5
Results using prompt-dependent grammars

Prompt type T_i	Sentence type U_i	WA	SU
Product order	Product order	93.39	94.36
Telephone number	Product order	0.1	0
Confirmation	Product order	−0.13	0
Telephone number	Telephone number	94.36	92.61
Post code	Telephone number	−37.3	0
Confirmation	Telephone number	−0.33	0
Post code	Post code	94.82	91.49
Address	Post code	−0.5	0
Confirmation	Post code	−0.15	0
Address	Address	96.3	85.66
Post code	Address	−0.03	0
Confirmation	Address	−0.21	0

and post codes because a digit wrongly recognized makes a telephone number or post code be wrongly understood, and analogously happens with the addresses: a wrongly recognized address item (e.g., street name, building floor, building number, etc.) makes the whole address be wrongly understood.

3.2.2. Contextual speech recognizer for in-context and out-of-context analysis

Secondly, we have analyzed the same four sentence types using a new input interface in which the standard HTK-based speech recognizer is substituted by the contextual speech recognizer proposed in this paper (Fig. 2). The contextual recognizer is comprised of the same HTK-based recognizer used in the previous experiment (but configured now to produce a word-network instead of a recognized sentence) and a WN-analyzer. No changes have been made in the semantic analyzer. The training corpus has been used to create the Ω set following the procedure described in Section 2.3, using the word-classes created for previous works. As result of the procedure we have obtained the sets S_i and the corresponding word-class bigrams C_i associated with the corresponding prompt types T_i the SAPLEN system can generate. Table 6 shows examples of sets S_i and word pair sequences associated with the prompt types T_i.

The new speech recognizer uses a word bigram compiled from the 4250 training sentences and produces a word-network for each input sentence. The WN-analyzer uses two parameters: prompt type (T_i) and probability increment (p). Using the first parameter we decide whether sentences are analyzed either in-context or out-of-context, whereas using the p parameter we decide how much the transition probabilities in the word-networks are incremented. If $p = 0$ the information concerning the word-class bigrams is not used and the WN-analyzer works just like a standard Viterbi recognizer (Rabiner and Juang, 1993) that analyzes sentences considering only acoustic and language probabilities, without considering the additional contextual information provided by the prompt-dependent bigrams.

3.2.2.1. Determination of the best value for the probability increment parameter. Since we want to compare the performance of the initial and the new input interfaces we must determine the best

Table 6
Examples of class-pair sets S_i and word-pair sequences associated with prompt types T_i

Prompt type T_i	Word-class pairs set S_i	S_i description	Word-pair sequences
T_1 = product order	S_1	(NUMBER, INDREDIENT)	one ham
		(INGREDIENT, FOOD)	ham sandwich
		(NUMBER, FOOD)	one sandwich
		(NUMBER, DRINK)	one milkshake
		(NUMBER, SIZE)	one small
		(NUMBER, TASTE)	three orange
		(TASTE, SIZE)	orange small
		(SIZE, TASTE)	small lemon
		(TASTE, TEMPERATURE)	lemon warm
		(SIZE, TEMPERATURE)	small cold
T_2 = telephone number	S_2	(NUMBER, NUMBER)	nine five
T_3 = post code	S_3	(NUMBER, NUMBER)	one eight
T_4 = address	S_4	(ADDRESS_NAME, ADDRESS_TYPE)	elm street
		(ADDRESS_TYPE, NUMBER)	street thirty
		(NUMBER, BUILDING_FLOOR)	thirty first
		(BUILDING_FLOOR, APPARTMENT_ LETTER)	first a

value of the probability increment and compare the results obtained for this value. To do so we assume that users answer the system prompts as expected most of the times, e.g., they utter a telephone number if the system prompts for a telephone number (in-context analysis), whereas they utter other sentence types occasionally, e.g., when they try to correct system errors (out-of-context analysis). Hence, it follows that the best value for the parameter must be determined in the in-context analysis. To find this value we have analyzed the sentences in the test corpus setting the T_i parameter match the sentence type being analyzed (in-context analysis), and have tested several values for the p parameter (0, 1, 2, 3, ...) until noticeable results have been obtained. The experiment has shown the lowest WA and SU scores are obtained when $p = 0$, the scores increase with the value of p until $p = 13$, and for greater values they decrease. So that, 13 has been found to be the best value for the probability increment parameter.

The experiment also shows that when $p < 13$ the WN-analyzer does not get enough benefit from the information provided by the word-class bigrams. This fact is easily observed from the trace files generated when the sentences are analyzed, as there are many word substitutions in the recognition process; for example, the word "sí" (yes) is often substituted by the words "seis" (six) or "sin" (without), the word "veintitrés" (23) is often substituted by the words "verde tres" (green three), the word "cero" (zero) is often substituted by the word "pero" (but), etc. These substitutions occur because the words sound very similarly in Spanish and the p parameter is set to a value that is not high enough to correct the wrong transitions. On the contrary, when $p > 13$ the WN-analyzer increments excessively the probability transitions causing a distortion in the analysis. In this case, the trace files show that there are many word insertions that follow the syntactic structure of the word-class sequences. Also, they show that meaningless words (that are not included in the word-classes) are often substituted by keywords (that are included in the word-classes); for

example, in many occasions the word "pero" (but) which is not a keyword is substituted by the word "queso" (cheese) which is a keyword.

3.2.2.2. In-context and out-of-context results for the best value of the probability increment. Once the best value for the probability increment parameter has been determined ($p = 13$), we have carried out experiments to observe the effect of using this value for the in-context and out-of-context analyses of the sentences in the test corpus, in order to compare the results with that obtained for the initial input interface. Table 7 shows the results obtained for the same cases of in-context and out-of-context analyses carried out with the initial interface (Section 3.2.1).

As can be observed, the results obtained for the out-of-context analysis are not excellent but are much better than those obtained for the initial input interface (Table 5). The average SU for the out-of-context case is 77.42%, which means that almost 8 out of 10 out-of-context analyzed sentences are correctly understood.

3.2.2.3. Linear interpolation of prompt-independent and prompt-dependent language models for sentence analysis. Finally, we have carried out experiments to compare the performance of the proposed procedure to re-score the word-networks based on the probability increment (p parameter) with that of using linear interpolation of the prompt-independent word bigram and the prompt-dependent word-class bigrams. The interpolation has been carried out as follows:

$$P(w_E \mid w_S) = (1 - \lambda)P_W(w_E \mid w_S) + \lambda P_{C_i}(w_E \mid w_S),$$

where P_W denotes the word bigram, P_{C_i} the word-class bigram mapped to the prompt T_i and λ is the interpolation weighting factor ($0 \leqslant \lambda \leqslant 1$). To carry out the sentence analysis we have modified the procedure to analyze the word-networks shown in Fig. 5: the p parameter is not used and thus the loop to decide the value of the 'increment' variable is by-passed. The transition language probability used ("p_L" parameter in Fig. 5) is not the one provided by the HTK-based recognizer ("l" parameter in Figs. 3 and 4) but that provided by the interpolated bigrams (in log format). Fig. 6 shows the average results obtained for the in-context and out-of-context analysis of the same sentence types considered in the previous sections.

Table 7
Results using the contextual speech recognizer for in-context and out-of-context sentence analysis ($p = 13$)

Prompt type T_i	Sentence type U_i	WA	SU
Product order	Product order	90.48	88.88
Telephone number	Product order	87.9	80.42
Confirmation	Product order	84.67	79.23
Telephone number	Telephone number	93.57	87.7
Post code	Telephone number	93.57	87.7
Confirmation	Telephone number	91.86	80.17
Post code	Post code	94.12	89.32
Address	Post code	91.4	77.79
Confirmation	Post code	91.62	77.9
Address	Address	92.49	84.6
Post code	Address	85.61	68.29
Confirmation	Address	84.62	67.9

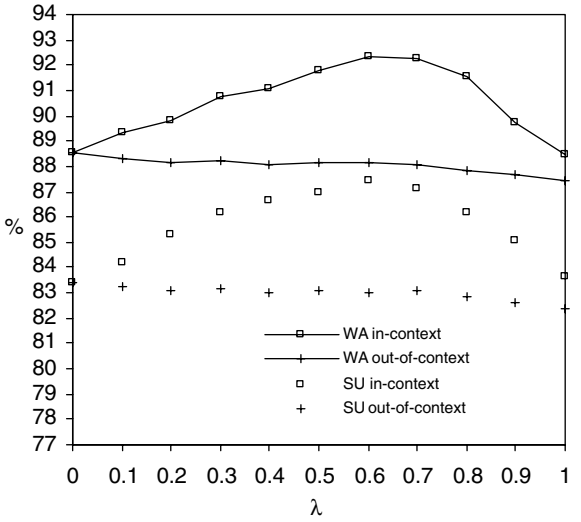

Fig. 6. Interpolation of word-bigram and word-class bigrams.

The figure shows the same effect in increasing the λ factor that we observed in previous experiments when increasing the p parameter. For the in-context analysis the performance enhances notably as λ increases until it reaches a threshold ($\lambda = 0.6$ for the sentence corpus used) obtaining absolute increments of 3.75% and 3.97% for WA and SU, respectively, in comparison with the case $\lambda = 0$. In the out-of-context analysis the performance deteriorates slightly as λ increases; the best scores are obtained for $\lambda = 0$ (WA = 88.57% and SU = 83.43%) while for the best value for the in-context analysis ($\lambda = 0.6$) the scores decrease to WA = 88.11% and SU = 83%.

If we compare the performance for the best values of both approaches ($p = 13$ vs $\lambda = 0.6$) we observe that the p parameter provides slightly higher scores for WA and SU: 92.66% and 87.62% vs 92.32% and 87.4% absolute, respectively. The increment in the scores for both measures is also slightly higher when the p parameter is used: 4.09% and 4.19% vs 3.75% and 3.97% absolute, respectively.

3.3. Limitations of the technique proposed in this paper

A major limitation of the technique proposed in this paper is that the pairs of word-classes can only capture the short-distance context dependency within a 2-word window. However, many of the context dependencies in natural language occur beyond such a window. Another disadvantage is that these pairs do not take into account the relations between particular words in word-classes, which can be very important for some languages in which words have gender and number correspondence relationships. For example, in our setting the WN-analyzer increases the probability of the transition "two sandwich" although the number correspondence is not observed. It makes sense to do so in our setting considering the south Spain accent of most users (as commented in Section 1) but it may be wrong to do so for standard Spanish or other languages. For these languages it would be preferable to use syntactic rules to check the gender/number correspondences between words before increasing transition probabilities. Finally, the technique does not

consider any information to filter out sentences that have no meaning in the domain. For example, in our setting the semantic of the sentence ''one chocolate cake'' is correct, but the sentences ''one apple sandwich'' and ''one ham cake'' have no meaning in the domain as these products do not exist in the system database. As the current setup does not take into account semantic information, the WN-analyzer increments the transition probabilities of these wrong sentences (in case they appear in a word-network).

4. Conclusions and future work

The experimental results show that the technique proposed in this paper enhances notably the performance of the new input interface of the SAPLEN system, which uses the contextual speech recognizer. The results also show that the information concerning word-classes is very important when analyzing the word-networks and that the value of the p parameter clearly influences the analysis. In comparison with a standard speech recognizer that uses a prompt-independent language model without context information (case $p = 0$), the contextual speech recognizer allows incrementing WA by 4.09% absolute, from 88.57 ($p = 0$) to 92.66% ($p = 13$), and SU by 4.19% absolute, from 83.43 ($p = 0$) to 87.62% ($p = 13$).

Comparing the performance of the initial and the new input interfaces, when the contextual speech recognizer is used WA increases by 93.71% absolute on average, from −4.81 (Table 5) to 88.90% (Table 7), and SU increases by 77.42% absolute on average, from 0 (Table 5) to 77.42% (Table 7); in other words, the proposed technique allows the system understand correctly approximately 8 out of 10 sentences out-of-context analyzed. The price to pay for this clear enhancement in the out-of-context analysis is a little reduction in the scores for the in-context analysis, as it can be observed when comparing the in-context results set out in both Tables. This comparison shows that when the contextual speech recognizer carries out the in-context analysis using $p = 13$, WA decreases by 2.05% absolute on average, from 94.71 (Table 5) to 92.66% (Table 7), and SU decreases by 3.41% absolute on average, from 91.03 (Table 5) to 87.62% (Table 7). These results indicate that if the users of the dialogue system would always answer the prompts with the appropriate sentence types, it would be preferable to use the initial input interface. However, in real dialogues users may utter sentences not permitted by the prompt-dependent language model, causing a system malfunction. Hence the proposed technique should be used to enhance the analysis of these sentences.

Very similar results have been obtained when analyzing the word-networks using linear interpolation of the prompt-independent language model (word bigram) and the prompt-dependent language models (word-class bigrams). The experiments show that both re-scoring methods work in a very similar way: WA and SU scores increase until reaching a threshold ($p = 13$, $\lambda = 0.6$), and for greater values they decrease. The scores obtained using both approaches are also very similar (slightly better if the p parameter is used) and the increment of the scores is also very similar (again slightly better in the case of the p parameter).

The future work includes studying alternative ways to enhance the procedure used to increment the transition probabilities in the word-networks. Using pairs of word-classes eases the procedure but enables that occasionally the probabilities of some transitions get wrongly incremented. As mentioned in Section 3.3, in order to avoid this problem it would be possible to include syntactic

and semantic rules to decide whether to increment probabilities; for example, a syntactic rule would suggest not to increment the probability of the transition "two → sandwich" because the number correspondence between both words is not observed, and a semantic rule would indicate not incrementing the probability of the transition "red → beer" because the product "red beer" does not have meaning in the application domain (as it does not exist in the product database of the system). However, semantic rules would be domain-dependent and then should be adapted if the system application is changed to deal with travel information, weather forecasts or other domains with different sentence types.

Acknowledgments

The authors thank the reviewers for their valuable comments to enhance this paper.

References

Baca, J.A., Zheng, F., Gao, H., Picone, J., 2003. Dialogue systems for automotive environments. In: Proc. Eurospeech, pp. 1929–1932.

Bernsen, N.O., 2003. One-line user modelling in a mobile spoken dialogue system. In: Proc. Eurospeech, pp. 737–740.

Bonneau-Maynard, H., Rosset, S., 2003. A semantic representation for spoken dialogs. In: Proc. Eurospeech, pp. 253–258.

Dahlbäck, N., Jönsson, A., Ahrenberg, L., 1993. Wizard of Oz studies – why and how. Proc. Int. Workshop on Intelligent User Interfaces, pp. 193–200.

Ehsani, F., Bernstein, J., Najmi, A., 2000. An interactive dialog system for leaning Japanese. Speech Communication 30, 167–177.

Emami, A., 2003. Improving a connectionist based syntactical language model. In: Proc. Eurospeech, pp. 413–416.

Fegyó, T., Mihajlik, P., Szarvas, M., Tatai, P., Tatai, G., 2003. Voxenter™ – Intelligent voice enabled call center for Hungarian. In: Proc. Eurospeech, pp. 1905–1908.

Filisko, E., Seneff, S., 2003. A context resolution server for the Galaxy conversational systems. In: Proc. Eurospeech, pp. 197–200.

Gaudinant, A., Goldman, J.-P., Wehrli, E., 1999. Syntax-based speech recognition: how a syntactic parser can help a recognition system. In: Proc. Eurospeech, pp. 1587–1590.

Hain, T., Woodland, P.C., Niesler, T.R., Whittaker, E.W.D. 1999. The 1998 HTK system for transcription of conversational telephone speech. In: Proc. International Conference on Acoustics, Speech and Signal Processing

Hardy, H., Baker, K., Bonneau-Maynard, H., Devillers, L., Rosset, S., Strzalkowski, T., 2003 Semantic and dialogic annotation for automated multilingual customer service. In: Proc. Eurospeech, pp. 201–204.

Heeman, P.A., Yang, F., Strayer, S.E., 2003. Control in task-oriented dialogues. In: Proc. Eurospeech, pp. 209–212.

Huang, Q., Cox, S., 2003. Automatic call-routing without transcriptions. In: Proc. Eurospeech, pp. 1909–1912.

Katz, S.M., 1987. Estimation of probabilities from sparse data for language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing 35 (3), 400–401.

Kawahara, T., Lee, C.H., Juang, B.H., 1998. Flexible speech understanding based on combined key-phrase detection and verification. IEEE Transactions on Speech and Audio Processing 6 (6), 558–568.

Kobayashi, N., Kobayashi, T., 1999. Class-combined word *n*-gram for robust language modeling. In: Proc. Eurospeech, pp. 1599–1602.

Lamel, L., Rosset, S., Gauvain, J.L., Bennacef, S., Garnier-Rizet, M., Prouts, B., 2000. The LIMSI Arise system. Speech Communication 31, 339–353.

Lane, I.R., Kawahara, T., Matsui, T., Nakamura, S., 2003. Hierarchical topic classification for dialog speech recognition based on language model switching. In: Proc. Eurospeech, pp. 429–432.

López-Cózar, R., Rubio, A.J., García, P., Segura, J.C., 1998. A spoken dialogue system based on a dialogue corpus analysis. In: Proc. First International Conference on Language Resources and Evaluation, pp. 55–58.

López-Cózar, R., Rubio, A.J., Díaz Verdejo, J.E., De la Torre, A., 2000. Evaluation of a dialogue system based on a generic model that combines robust speech understanding and mixed-initiative control. In: Proc. Language Resources and Evaluation Conference, pp. 743–748.

López-Cózar, R., Milone, D.H., 2001. A new technique based on augmented language models to improve the performance of spoken dialogue systems. In: Proc. Eurospeech, pp. 741–744.

López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A.J., López-Soler, J.M., 2002. A new method for testing dialogue systems based on simulations of real-world conditions. In: Proc. International Conference on Speech and Language Processing, pp. 305–308.

López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A.J., 2003. Assessment of dialogue systems by means of a new simulation technique. Speech Communication 40 (3), 387–407.

Mori, S., Nishimura, M., Itoh, N., 2003. Language model adaptation using word clustering. In: Proc. Eurospeech, pp. 425–428.

Nakano, N., Minami, Y., Seneff, S., Hazen, T.J., Cyphers, D.S., Glass, J., Polifroni, J., Zue, V., 2001. Mokusei: A telephone-based Japanese conversational system in the weather domain. In: Proc. Eurospeech, pp. 1331–1334.

Niesler, T.R., Woodland, P.C., 1999. Variable-length category *n*-gram language models. Computer, Speech and Language 13 (1), 99–124.

Niimi, Y., Tomoki, O., Nishimoto, T., Araki, M., 2000. A task-independent dialogue controller based on the extended frame-driven method. In: Proc. International Conference on Speech and Language Processing, pp. 114–117.

Pellom, B., Ward, W., Pradham, S., 2000. The CU communicator: an architecture for dialogue systems. In: Proc. International Conference on Speech and Language Processing, pp. 723–726.

Rabiner, L.R., Juang, B.H., 1993. Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

Seneff, S., Polifroni, J. 2000. Dialogue management in the Mercury flight reservation system. In: Proc. ANLP-NAACL Workshop on Conversational Systems, Seattle, Washington.

Seto, S., Kanazawa, H., Sinchi, H., Takebayashi, Y., 1994. Spontaneous speech dialogue system TOSBURG II and its evaluation. Speech Communication 15, 341–353.

Siu, M., Ostendorf, M., 2000. Variable *n*-grams and extensions for conversational speech language modeling. IEEE Transactions on Speech and Audio Processing 8, 63–75.

Takeuchi, M., Kitaoka, N., Nakagawa, S., 2003. Generation of natural response timing using decision tree based on prosodic and linguistic information. In: Proc. Eurospeech, pp. 613–616.

Visweswariah, W., Prints, H., 2001. Language models conditioned on dialog state. In: Proc. Eurospeech, pp. 251–254.

Wang, C., Cyphers, S., Mou, X., Polifroni, J., Seneff, S., Yi, J., Zue, V., 2000. MuXing: a telephone-access Mandarin conversational system in the weather domain. In: Proc. International Conference on Speech and Language Processing, pp. 715–718.

Wang, H.-M., Lin, Y.-C., 2003. Sentence verification in spoken dialogue system. In: Proc. Eurospeech, pp. 625–628.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P. 2000. The HTK Book (for HTK Version 3.0). Microsoft Corporation.

Zhang, X., Huang, C., Zhao, S., Huang, T. 1998. Spoken language understanding in spoken dialogue system for travel information accessing. In: Proc. International Conference on Speech and Language Processing, pp. 294–298.

Zitouni, I., Siohan, O., Lee, C.-H., 2003. Hierarchical class *n*-gram language models: towards better estimation of unseen events in speech recognition. In: Proc. Eurospeech, pp. 237–240.

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