New Technique to Enhance the Performance of Spoken Dialogue Systems Based on Dialogue States-Dependent Language Models and Grammatical Rules

Ramón López-Cózar[1], *David Griol*[2]

[1]Dept. of Languages and Computer Systems, University of Granada, Spain
[2]Dept. of Computer Science, Carlos III University of Madrid, Spain
rlopezc@ugr.es ; dgriol@inf.uc3m.es

Abstract

This paper proposes a new technique to enhance the performance of spoken dialogue systems which presents one novel contribution: the automatic correction of some ASR errors by using language models dependent on dialogue states, in conjunction with grammatical rules. These models are optimally selected by computing similarity scores between patterns obtained from uttered sentences and patterns learnt during training. Experimental results with a spoken dialogue system designed for the fast food domain show that our technique allows enhancing word accuracy, speech understanding and task completion rates of a spoken dialogue system by 8.5%, 16.54% and 44.17% absolute, respectively.
Index Terms: Spoken dialogue systems, language modelling, speech recognition.

1. Introduction

Most techniques available in the literature to correct ASR errors employ statistical knowledge about uttered and recognised words [1] [2]. A problem with these techniques is that they need vast amounts of training data. Moreover, their success depends on the quality of the ASR output and on the size of the database of errors used for training. To address these problems, several authors have used lexical, syntactic or semantic information, and some of them have employed knowledge concerned with dialogue management [3] [4]. The technique that we propose considers statistical information and several information sources to correct ASR errors, namely lexical, syntactic, semantic and dialogue-related. The main novelty is that it takes into account prompt-dependent models to correct the errors, being the optimal model selected by the computation of a similarity score between the pattern obtained from the uttered sentence and patterns learnt during training. In addition, our technique considers grammatical rules to correct errors that cannot be detected using these models.

2. Elements to implement the technique

2.1. Concepts

We define a *concept* as a set of keywords of a given type which are necessary to extract the semantic content of sentences within an application domain. For example, in our experiments in the fast food domain, we consider, among others, the following concepts: DESIRE = {want, need, …}, FOOD = {sandwich, cake, salad, …}, DRINK = {water, beer, wine, …} and AMOUNT = {one, two, three, …}.

2.2. Grammatical rules

The general format of a grammatical rule is as follows: *ssp* → *restriction*, where *ssp* denotes a syntactic-semantic pattern, which will be described in the following section, and *restriction* is a condition that must be satisfied by all the concepts in the pattern. For example, one rule used in our experiments is:

NUMBER DRINK SIZE →
number(NUMBER) = *number*(DRINK) **and**
number(DRINK) = *number*(SIZE) **and**
number(NUMBER) = *number*(SIZE)

where *number* is a function that returns either 'singular' or 'plural' for each word in the concepts that it uses as input. The goal of this rule is to check number correspondences of drink orders uttered in Spanish. For example, the sentence "dos cervezas grandes" (two large beers) holds this correspondence.

2.3. Syntactic-semantic models

A syntactic-semantic model is a conceptual representation of the sentences uttered by users of a spoken dialogue system (SDS) in a dialogue state T. This state is associated with a prompt type of the system, which represents equivalent prompts to obtain a particular data from the user. To create a syntactic-semantic model for a dialogue state T, we transform each sentence uttered in a dialogue state into what we call a *syntactic-semantic pattern* (*ssp*). This pattern is a sequence of concepts obtained by replacing each word in the sentence with the concept(s) the word belongs to. From the analysis of all the sentences uttered in response to each prompt type we create a set of *ssp*'s, in which we remove those that are redundant and associate to each *ssp* its relative frequency within the set. The outcome of this process is a syntactic-semantic model associated with the prompt type T (SSM_T). We call α model the set of SSM_T's created considering the m prompt types of a SDS: $\alpha = \{SSM_{Ti}\}$, $i = 1 \ldots m$.

2.4. Lexical models

The lexical models contain information about the performance of the speech recogniser of a SDS. We must create a lexical model for each dialogue state T, which we call LM_T. To do so, we consider the sentences uttered in the dialogue state and their corresponding recognition results. The format of this model is: $LM_T = \{w_a, w_b, p_{ab}\}$, where w_a is a word uttered by a user, w_b is the recognised word and p_{ab} is

This research has been funded by Spanish project HADA TIN2007-64718.

the posterior probability of obtaining w_b given w_a. To create LM_T we align each uttered sentence with the recognised sentence using the method described in [5], and compute the probabilities p_{ab} for each word pair (w_a, w_b). We call *β model* the set of LM_T's created considering the *m* prompt types of a SDS: $\beta = \{LM_{Ti}\}$, $i = 1 \ldots m$.

2.5. Algorithms to implement the technique

2.5.1. Correction at statistical level

The goal of this correction level is to find words w_I's in the recognised sentence which belong to incorrect concepts K_I's. For each word, we must decide the correct concept K_C and select the most appropriate word $w_C \in K_C$ to substitute w_I in the recognised sentence. We can implement this procedure in two steps:

Step 1. Pattern matching. This step employs what we call an *enriched syntactic-semantic pattern* ($essp_{INPUT}$) obtained from the recognised sentence. This pattern is a sequence of what we call *containers*. The goal of this step is to transform $essp_{INPUT}$ into another pattern called $essp_{BEST}$, which is initially empty. To create this new pattern, we firstly create a syntactic-semantic pattern called ssp_{INPUT}, which only contains the concepts in $essp_{INPUT}$, for example: ssp_{INPUT} = DESIRE AMOUNT INGREDIENT FOOD.

Next, we decide whether ssp_{INPUT} matches any pattern in the syntactic-semantic model associated with the dialogue state T (SSM_T). If so, we make $essp_{BEST} = essp_{INPUT}$ and proceed with the correction at the linguistic level (section 2.5.2). Otherwise, we look for patterns similar to ssp_{INPUT} in SSM_T. To do this we compare ssp_{INPUT} with every pattern *p* in the model, and compute a similarity score as follows: $similarity(ssp_{INPUT}, p) = (n - m_{ed}) / n$, where *n* is the number of concepts in ssp_{INPUT} and m_{ed} is the minimum edit distance between both patterns, computed using the method described in [6]. We call $ssp_{SIMILAR}$ any pattern *p* in SSM_T such that $similarity(ssp_{INPUT}, p) > t$, where $t \in [0.0, 1.0]$ is a similarity threshold, the optimal value of which must be experimentally determined. We consider 3 cases depending on the number of $ssp_{SIMILAR}$'s in SSM_T:

Case 1. There is just one $ssp_{SIMILAR}$ in SSM_T. Thus, we create a new pattern called ssp_{BEST}, make $ssp_{BEST} = ssp_{SIMILAR}$ and proceed with Step 2 (Pattern alignment).

Case 2. There are no $ssp_{SIMILAR}$'s in SSM_T. Thus, we try to find $ssp_{SIMILAR}$'s in the α model (discussed in section 2.3). If no $ssp_{SIMILAR}$'s are found, we do not make any correction at the statistical level; if there is just one, we proceed as in Case 1; if there are several, we proceed as in Case 3.

Case 3. There are several $ssp_{SIMILAR}$'s in SSM_T (or in α). The question then is to decide the best $ssp_{SIMILAR}$. To make this selection we search for the $ssp_{SIMILAR}$ that has the greatest similarity with ssp_{INPUT}. If there is just one $ssp_{SIMILAR}$ satisfying this condition, we make $ssp_{BEST} = ssp_{SIMILAR}$ and proceed with Step 2. If there are several patterns, we select those with the highest frequency in SSM_T (or in α): if there is just one, we make $ssp_{BEST} = ssp_{SIMILAR}$ and proceed with Step 2; if there are several we do not make any correction at the statistical level.

Step 2. Pattern alignment. The goal of this step is to build $essp_{BEST}$ in case it is still empty. To do this, we take into account each container C_a in ssp_{INPUT} and consider three cases:

Case A. The word w_a in C_a does not affect the semantics of the sentence, i.e., it is not a keyword (e.g. '*please*'). Thus, we create a new container *D*, make $D = C_a$ and add *D* to $essp_{BEST}$.

Case B. The word w_a in C_a affects the semantics of the sentence, i.e., it is a keyword (e.g. '*sandwich*'). Thus, we study whether the word must be corrected. To do this, we try to align the container C_a with a container C_b in ssp_{BEST} using the method described in [5] and consider 3 cases:

Case B.1. C_a can be aligned. In this case we assume that the container C_a is correct and do not make any correction at the statistical level. We create a new container *D*, make $D = C_a$ and add *D* to $essp_{BEST}$.

Case B.2. It is not possible to align C_a. This case may happen in the two following situations:

Case B.2.1. The container is a result of an insertion recognition error. In this case we discard C_a, i.e. it is not added to $essp_{BEST}$.

Case B.2.2. The container is a result of a substitution recognition error. Therefore, we must find a correction word from a different concept, $w_C \in C_b$, store it in a new container *D*, and add this container to $essp_{BEST}$. To find w_C we consider the lexical model associated with the dialogue state T (LM_T) and create the set *U* of words $u \in C_b$ with which the word w_I is confused. If there is only one word *u* in *U*, we create a new container *D* that we name C_b, store it in *u*, and add *D* to $essp_{BEST}$. If there are several words, we carry out the same procedure but using the word that has the highest confusion probability with w_I if it is unique; if it is not unique, or there are no words in *U*, we do not make any correction at the statistical level.

2.5.2. Correction at the linguistic level

The goal of this correction level is to repair errors that are not detected at the statistical level and which affect the semantics of the sentences. To carry out the correction we use the grammatical rules described in section 2.2. For each rule we carry out the following procedure. The syntactic-semantic pattern *ssp* of the rule is inserted in a *window* that slides from left to right over $essp_{BEST}$, as can be observed in Fig. 1.

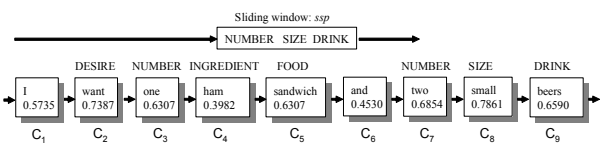

Fig. 1. Sliding window over $essp_{BEST}$.

If the concept sequence in the window is found in $essp_{BEST}$, then we apply the *restriction* of the rule to the words in the containers of $essp_{BEST}$. If the words satisfy the restriction, we do not make any correction. Otherwise, we try to find out the reason for the insatisfaction by searching for an incorrect word w_I. To decide the word w_C to correct the incorrect word, we consider the lexical model LM_T and take into account the set $U = \{u_1, u_2, \ldots, u_p\}$ comprised of words of the same

concept than the word w_I. Next, we proceed similarly as discussed in Case B.2.2 but considering that the goal now is to replace one word in one concept with other word in the same concept.

3. Experiments

The goal of the experiments is to test the proposed technique using the Saplen system, which we developed in a previous study to answer fast food queries and orders made in Spanish [7]. The evaluation has been carried out in terms of word accuracy (WA), speech understanding (SU) and task completion (TC), considering two front-ends for ASR: i) *baseline ASR*, comprised of the standard HTK-based speech recogniser of the Saplen system, and ii) *enhanced ASR*, comprised of the same speech recogniser plus an additional module that implements the proposed technique.

We have employed a dialogue corpus collected in our University from students interacting with the Saplen system, which contains around 5,500 utterances and roughly 2,000 different words. The utterance corpus has been divided into two separate corpora, each containing around 50% of the utterances. Using the training corpus we have compiled a word bigram that allows recognising sentences of the 18 different types in the corpus. The remaining 50% of the utterances have been used for testing.

The experiments have been carried out employing a user simulator developed in a previous study [8]. The interaction between the Saplen system and the simulator is decided considering a set of scenarios that represent user goals. We have created two scenario sets: *ScenariosA* (300 scenarios) and *ScenariosB* (100 scenarios). Each dialogue generated by the interaction between the Saplen system and the user simulator is stored in a log file for analysis and evaluation purposes.

Given that the construction of the syntactic-semantic and lexical models described in sections 2.3 and 2.4 has been carried out employing simulated dialogues, we have made additional experiments to decide the necessary number of dialogues to obtain the maximum amount of syntactic-semantic and lexical knowledge. The results indicate that 900 dialogues is the optimal trade-off.

3.1. Experiments with the baseline ASR

Employing the user simulator, the Saplen system and *ScenariosA,* we have generated a corpus of 900 dialogues, which we have called $DialoguesA_1$. Table 1 sets out the average results obtained from the analysis of this corpus. The results show the problems of the system in correctly recognising and understanding some utterances. Analysis of the log files reveals that in some cases the misrecognised sentences are similar to the uttered sentences. For example, "dos fantas grandes de limón" (two large lemon fantas) is recognised as "uno fantas grandes de limón" (one large lemon fantas) because of the acoustic similarly between 'dos' and 'uno' when uttered by users with strong Southern Spanish accents.

Table 1. *Results using the baseline ASR (in %).*

WA	SU	TC
76,12	54,71	24,51

We have also observed problems with confirmations, which happen because the speech recogniser usually substitutes the word 'sí' (yes) by the word 'seis' (six), when the former word is uttered by strongly accented speakers. In other cases, the recognised sentences are very distorted by ASR errors. For example, the sentence "quiero una fanta de naranja grande" (I want one big orange Fanta) is sometimes recognised as "queso de manzana tercera" (cheese of apple third).

3.2. Experiments with the enhanced ASR

As the *concepts* required for the technique (discussed in section 2.1), we have employed a set of 21 *concepts* that we created in a previous study [7]. Following section 2.2 we have created a set of grammatical rules to check the number correspondences for food and drink orders. To create the syntactic-semantic and lexical models, discussed in sections 2.3 and 2.4, we have analysed $DialoguesA_1$ thus obtaining $\alpha = \{SSM_{Ti}\}$ and $\beta = \{LM_{Ti}\}$, with i = 1 ... 43 given that the Saplen system can be in 43 different dialogue states.

To decide the optimal value for the similarity threshold t (discussed in section 2.5.1) we have carried out experiments considering values in the range [0.1, 0.9]. Employing the user simulator and *ScenariosB*, we have generated a corpus comprised of 300 dialogues for each value, using in all cases the proposed technique. Analysis of the outcomes of these experiments reveals that the best results are obtained when t = 0.5. Using this optimal value, we have employed again *ScenariosA* to generate another corpus of 900 dialogues, which we call $DialoguesA_2$. Table 2 shows the average results obtained from the analysis of this corpus.

Table 2. *Results using the enhanced ASR (in %).*

WA	SU	TC
84,62	71,25	68,32

Analysis of the log files shows that the technique is successful in correcting some incorrectly recognised sentences. For example, the incorrectly recognised drink order "one large lemon fantas" is corrected by doing no changes at the syntactic-semantic level, and replacing 'one' with 'two' at the lexical level. In other product orders the correction is carried out at the semantic-syntactic level. For example, "one curry salad" is sometimes recognised as "one error curry salad". In this case the correction is carried out removing the ERROR concept at the syntactic-semantic level.

The technique is useful in correcting the errors with confirmations discussed in the previous section. To do this, it replaces the NUMBER concept with the CONFIRMATION concept, and then selects the most likely word in CONFIRMATION.

The enhanced ASR enables as well correction of some misrecognised telephone numbers. For example, "nine five eight twenty-one fourteen eighteen" is sometimes recognised as "gimme five eight twenty-one fourteen eighteen" because of acoustic similarity between 'nine' and 'gimme' in Spanish. The technique corrects the error by replacing the DESIRE concept with the NUMBER concept and selecting the most likely word in NUMBER given the word 'gimme' at the lexical level.

The technique is also useful to correct some misrecognised postal codes. For example, "eighteen zero zero one" is sometimes recognised as "eighteen zero zero turkey". This error is corrected by replacing the INGREDIENT concept with the NUMBER concept and selecting the most likely word in NUMBER given the word 'turkey'.

Our proposal is also successful in correcting some incorrectly recognised addresses (in the Spanish format). For example, "almona del boquerón street number five second

floor letter h" is sometimes recognised as "almona del boquerón street error five second floor letter zero". This error is corrected by making a double correction. First, replacement of the ERROR concept with the NUMBER_ID concept and selection of the most likely word in NUMBER_ID given the word 'error'. Second, replacement of the NUMBER concept with the LETTER concept and selection of the most likely word in LETTER given the word 'zero'.

There are cases where the technique fails in detecting errors, and thus in correcting them. This happens when words in the uttered sentence are substituted by other words and the result is valid in the application domain. For example, this occurs when the sentence "two green salads" is recognised as "twelve green salads", given that there is no conflict in terms of *concepts* and there is agreement in number between the words.

3.2.1. Advantage of using SSM_T's, α and t

In this experiment we have checked whether using SSM_T's or α, taking into account *t*, is preferable to the two following alternative strategies: i) use α only without firstly checking the SSM_T's, and ii) use the SSM_T's, but if the pattern ssp_{INPUT} is not found in these, use α without considering the similarity threshold *t*. The α model is the one created employing $DialoguesA_1$ and *t* is set to the optimal value, i.e., $t = 0.5$. We have implemented strategy i) and used *ScenariosA* to generate a corpus of 900 dialogues, which we call $DialoguesA_3$. Next, we have implement strategy ii) and, using again *ScenariosA*, have generated another corpus of 900 dialogues, which we call $DialoguesA_4$. Therefore, $DialoguesA_1$, $DialoguesA_3$ and $DialoguesA_4$ have been created using the same scenarios and are comprised of the same number of dialogues, the only difference being in the strategy for selecting the correction model to be used. Table 3 shows the average results obtained from the analysis of $DialoguesA_3$ and $DialoguesA_4$.

Table 3. *Results employing alternative strategies to select the syntactic-semantic correction model (in %).*

Corpus	WA	SU	TC
$DialoguesA_3$	80.15	61.67	39.78
$DialoguesA_4$	82.26	66.84	55.35

Analysis of the log files shows that the error correction in confirmations is very much affected by the strategy employed to select the correction model (either SSM_T or α). If we always use SSM_T to correct errors in confirmations, the correction is in many cases successful. On the other hand, if we always use α the correction is mostly incorrect.

3.2.2. Advantage of using LM_T's, β and t

The goal of this experiment has been to check whether using the LM_T's or β taking into account *t* is preferable to using β regardless of *t*. To carry out the experiment we have used the β model created with $DialoguesA_1$. We have employed again *ScenariosA* and generated a corpus of 900 dialogues, which we call $DialoguesA_5$. Therefore, $DialoguesA_1$ and $DialoguesA_5$ have been obtained using the same scenarios and are comprised of the same number of dialogues, the only difference being in the use of β. Table 4 shows the average results obtained from the analysis of $DialoguesA_5$. The experiment shows that the confusion probabilities of words are not the same in the LM_T's and β. For example, considering the β model, the highest probability of confusing the word 'error' with a word in the NUMBER concept is 0.0370, and this word is 'dieciseis' (sixteen). However, considering $LM_{T=PRODUCT\text{-}ORDER}$, this probability is 0.0090 and the word is 'una' (one). Therefore, the correction word is 'dieciseis' if we consider β, and 'una' if we take into account $LM_{T=PRODUCT\text{-}ORDER}$, which in some cases is deterministic in making the proper correction.

Table 4. *Results employing an alternative strategy to select the lexical model (in %).*

Corpus	WA	SU	TC
$DialoguesA_5$	81.40	65.61	60.89

4. Conclusions and future work

Comparing the results set out in Tables 1 and 2 we observe that the proposed technique allows enhancing the performance of the Saplen system in terms of WA, SU and TC by 8.5%, 16.54% and 44.17% absolute, respectively. These enhancements are mostly achieved because considering the proposed threshold for similarity scores between patterns, the technique decides whether to use correction models associated with the current dialogue state (SSM_T and LM_T), or general correction models for the application domain (α and β). This novel contribution optimises the procedure for error recovery, as can be observed from comparison of results set out in Tables 2, 3 and 4. These results show that our method for selecting the correction models is preferable to other possible strategies for selecting these models. In particular, we have observed that the benefit of the proposed method is particularly noticeable in the correction of misrecognised confirmations.

Future work includes considering additional information sources to correct errors that in the current implementation cannot be detected, such as domain-dependent knowledge. For example, in our application domain we could use this kind of information to consider that the sentence "twelve green salads", although syntactically correct, is likely to be incorrectly recognised, given that it is not usual that the users order such a large amount of a product. We also plan to study the performance of the technique considering prompt-dependent similarity thresholds.

5. References

[1] E. K. Ringger, J. F. Allen, "A fertility model for post correction of continuous speech recognition", Proc. ANLP-NAACL Satellite Workshop, pp. 1-6, 2000.

[2] Z. Zhou, H. Meng, W. K. Lo, "A multi-pass error detection and correction framework for Mandarin LVCSR", Proc. ICSLP, pp. 1646-1649, 2006.

[3] M. Jeong, B. Kim, G. G. Lee, "Semantic-oriented correction for spoken query processing", Proc. of ASRU, pp. 156-161, 2003.

[4] S. Jung, M. Jeong, G. G. Lee, "Speech recognition error correction using maximum entropy language model", Proc. Interspeech, pp. 2137-2140, 2004

[5] W. M. Fisher, J. G. Fiscus, "Better alignment procedures for speech recognition evaluation", Proc. ICASSP, pp. 59-62, 1993.

[6] F. Crestani, "Word recognition errors and relevance feedback in spoken query processing", Proc. Conf. on Flexible Query Answering Systems, pp. 267-281, 2000.

[7] R. López-Cózar, Z. Callejas, "Combining language models in the input interface of a spoken dialogue system", Computer Speech and Language, 20, pp. 420-440, 2006.

[8] R. López-Cózar, A. de la Torre, J. C. Segura, A. J. Rubio, V. Sánchez, "Assessment of dialogue systems by means of a new simulation technique", Speech Communication, 40(3), pp. 387-407, 2003.