New ASR Technique to Enhance the Performance of Spoken Dialogue Systems

R. López-Cózar[1], *Gonzalo Espejo*[1], *Nieves Ábalos*[1], *David Griol*[2], *José F. Quesada*[3]

[1] Dept. of Languages and Computer Systems, CITIC-UGR, University of Granada, Spain
[2] Dept. of Computer Science, Carlos III University of Madrid, Spain
[3] Dept. of Artificial Intelligence and Computer Science, Seville, Spain

rlopezc@ugr.es, {gonzaep,nayade}@correo.ugr.es, dgriol@inf.uc3m.es, Jose.Quesada@infinity.es

Abstract

In this paper we present a new technique to enhance the performance of spoken dialogue system, which employs contextual models and grammatical rules to optimise the automatic correction of some errors made by the ASR component of these systems. Different experiments have been carried out to evaluate this technique employing a previously developed spoken dialogue system designed for the fast food domain. The results of this experimentation show that our technique enhances the performance of the system, by notably incrementing the rates of word accuracy, speech understanding and task completion.

Index Terms: Spoken Dialogue Systems, Automatic Speech Recognition, Language Modelling.

1. Introduction

Spoken dialogue systems (SDSs) are employed nowadays by many companies to provide automatic services such as travel information [1, 2], weather forecasts [3, 4], fast food ordering [5, 6], call routing [7] or directory assistance [8]. These systems can be very convenient for companies as they enable important economic savings in providing services available 24 hours a day, 365 days a year. In addition, these systems can be very handy for users, as they can get easily information (in theory) by means of spontaneous speech using a telephone. However, many users reject using these systems because the interaction many times is not very natural and friendly.

The clearly observable differences with respect to human-to-human interaction are caused by several reasons. One is the current limitations of state-of-the-art automatic speech recognition (ASR) for real-world applications. Hence, to make these systems more widely accepted by all potential users, it is very important to develop methods to increase the robustness of the speech recogniser. One method to do this is by automatically correcting some errors made by this system's component.

Many techniques can be found in the literature addressing this task. For example, [9] proposed to use a channel model and a language model, in which the former takes into account errors made by a speech recogniser whereas the latter provides information about sequences of uttered words. Following a different approach, [10] proposed a technique that carries out ASR correction at two levels of analysis. The former uses a classifier to decide whether the outcome of the ASR is incorrect; if it is, the outcome is passed on to the second level of analysis, where another classifier is used to decide the incorrect words.

One problem with the techniques described above is that they rely on statistical information only, and thus need vast amounts of training data. To overcome this drawback a number of authors have proposed to combine lexical, syntactic or semantic information, and some of them have employed knowledge concerned with dialogue management [11]. Following this approach, the technique that we propose considers statistical, lexical, syntactic, semantic and dialogue-related information. 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.

After this short introduction, the remainder of the paper is organised as follows. Section 2 presents the proposed technique, discussing the information required and the algorithms for implementation. Section 3 addresses the experiments, in which we compare results on word accuracy, sentence understanding and task completion, with and without using the proposed technique in a SDS that we developed in a previous study.

2. The proposed ASR technique

The proposed ASR technique to enhance the performance of SDSs is based on the use of semantic, grammatical and lexical information at the ASR level, as described in the following sections.

2.1. Semantic information

The semantic information is represented by means of what we call *semantic classes*. A semantic class is 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 semantic classes: DESIRE = {want, need, …}, FOOD = {sandwich, cake, salad, …}, DRINK = {water, beer, wine, …} and AMOUNT = {one, two, three, …}.

2.2. Grammatical information

The grammatical information is represented by means of rules of the following form: *ssp* → *restriction*, where *ssp* denotes a syntactic-semantic pattern and *restriction* is a condition that

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

must be satisfied by all the semantic classes in the pattern. For example, one rule used in our experiments is as follows:

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 semantic classes NUMBER, DRINK and SIZE. 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.

We consider that a dialogue state T represents a prompt type of a SDS by means of which the system expects to obtain a particular type of data from the user, for example, a telephone number. We consider as well that the sentences uttered by users in a dialogue state T can be represented by what we call a *syntactic-semantic* model. To create such a model, we transform each sentence into what we call a *syntactic-semantic pattern* (*ssp*). This pattern is a sequence of semantic classes obtained by replacing each word in the sentence with the semantic class(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:

$$\alpha = \{SSM_{Ti}\},\ i = 1 \ldots m.$$

2.3. Lexical information

The lexical information takes into account the performance of the speech recogniser of a SDS. In accordance with our technique, 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 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 [12], 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:

$$\beta = \{LM_{Ti}\},\ i = 1 \ldots m.$$

2.4. Algorithms to implement the technique

The correction of ASR errors is performed at two levels (statistical and linguistic) as explained in the following sections.

2.4.1. 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 semantic classes K_I's. For each word, we must decide the correct semantic class 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*. Each container stores a word of the sentence and has a name if the word is a keyword, which is the name of the semantic class the word belongs to (e.g., DESIRE). 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 semantic classes 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 2.4.2). 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 semantic classes in ssp_{INPUT} and m_{ed} is the minimum edit distance between both patterns, computed using the method described in [13]. 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 2.2). 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 [12] and consider three 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 semantic class, $w_C \in C_b$, 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.

2.4.2. 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 $essp_{BEST}$. If the sequence of semantic classes 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 $U = \{u_1, u_2, ..., u_p\}$ comprised of words of the same semantic class 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 semantic class with other word in the same semantic class.

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 [6]. 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 [15]. 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 2.2 and 2.3 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 semantic classes required for the technique (discussed in section 2.1), we have employed a set of 21 semantic classes that we created in a previous study [14]. 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 2.2 and 2.3, 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 2.4.1) 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 semantic class 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 semantic class NUMBER with the semantic class CONFIRMATION, 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 semantic class DESIRE with the semantic class NUMBER 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 semantic class INGREDIENT with the semantic class NUMBER 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 semantic class ERROR with the semantic class NUMBER_ID and selection of the most likely word in NUMBER_ID given the word 'error'. Second, replacement of the semantic class NUMBER concept with the semantic class LETTER 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 semantic classes and there is agreement in number between the words.

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, or general correction models for the application domain. 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] Seneff S., Polifroni J. "Dialogue management in the Mercury flight reservation system", Proc. of ANLP-NAACL Satellite Workshop, pp. 1-6, 2000.

[2] Billi, R., Castagneri, G., Danielli, M. "Field trial evaluations of two different information inquiry systems", *Speech Communication*, 23, pp. 83–93, 1997.

[3] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L. "Jupiter: A telephone-based conversational interface for weather information",. *IEEE Trans. on Speech and Audio Proc.*, 8(1), pp. 85-96, 2000.

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

[5] Seto, S., Kanazawa, H., Shinchi, H., Takebayashi, Y. "Spontaneous speech dialogue system TOSBURG II and its evaluation", *Speech Communication*, 15, pp. 341-353, 1994.

[6] López-Cózar, R., García, P., Díaz, J., Rubio, A. J. "A voice activated dialogue system for fast-food restaurant applications", Proc. of Eurospeech, pp. 1783-1786, 1997.

[7] Lee, C.-H., Carpenter, B., Chou, W., Chu-Carroll, J., Reichl, W., Saad, A., Zhou, Q. "On natural language call routing", *Speech Communication*, 31, pp. 309-320, 2000.

[8] Kellner, A., Rueber, B., Seide, F., Tran, B.-H. "PADIS – An automatic telephone switchboard and directory information system", *Speech Communication*, 23, pp. 95–111, 1997.

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

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

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

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

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

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

[15] López-Cózar, R., Callejas, Z., McTear, M. "Testing the performance of a spoken dialogue system by means of a new artificially simulated user", Artificial Intelligence Review, 26, pp. 291-323, 2006