

Using Dialog Corrections to Improve Speech Recognition

Marco Orlandi[†], Christopher Culy[‡], Horacio Franco[‡]

[†]ITC-irst, Pantè di Povo, Trento, Italy

orlandi@itc.it

[‡]SRI International, Menlo Park, CA, USA

culy@ai.sri.com, hef@speech.sri.com

Abstract

We propose a preliminary method for automatically correcting errors in spoken dialogue systems¹. Current spoken dialogue systems usually show a rather static and rigid behavior regarding recognition errors, therefore a feasible method of correcting system errors might be helpful to successfully support user requests. Moreover, a correction differs from non-correction prosodically [1]. Generally a user correction exhibits a greater prosodic difference the more distant it is from the initial error. In this case it is recognized more poorly, and it involves a longer human-machine interaction because this often leads to the same recognition errors.

This paper will focus on methods to adapt the system using the error feedback provided by the user, and basically requires the adaptation of the language model.

1. Introduction

At present, dialogue technology is sufficiently robust for handling information access in restricted domains (e.g. train timetable inquiry) and in larger domains (e.g. requests of tourism information) [2]. Furthermore, the portability of the technology towards different domains is guaranteed by an application independent architecture and by easy to use development tools and interfaces [3]. For making the system behavior less static and rigid, with respect to recognition errors, some new techniques need to be developed. In fact, a recurrent error can cause problems for a dialogue system, forcing the user to repeat information in more turns, decreasing the dialogue efficiency and even resulting in a complete failure [4]. In this paper, we propose an error correction mechanism that allows to improve the robustness of dialogue systems by detecting and recovering errors arising from speech recognition. This method requires to adapt the language model using the error feedback provided by the user.

At the beginning of the work a “spoken address” domain was chosen, composed of a few basic semantic ele-

ments (i.e. street number, street name and city are mandatory; zip code and state are optional). An earlier system was developed in SRI Labs, but the basic concepts can be transferred towards other applications. We collected data and studied how users correct recognition errors during an interaction with the machine. This data collection and this preliminary system have been useful for understanding the goodness of our approach and to demonstrate the effectiveness of the technology. Since the performance obtained on this latter task was quite satisfactory, a new prototype, working on dates in Italian, has been recently developed.

In this paper we define the problem, explain the basic ideas and report the early steps we have done to achieve the goal.

2. System Description

A software architecture for information access in restricted domains is depicted in Fig. 1. It is formed by various modules, namely: dialogue engine, automatic speech recognizer, grammar builder, acoustic database and graphical user interface.

The Dialogue Manager controls the spoken interaction flow according to a logic contained in a description written with a specific language. The dialogue engine has to interpret the description of an application, which is both declarative (for what concerns contexts and concepts) and procedural (for the definition of the actions that must be executed in some dialogue states). Each concept has associated with it a set of features, that specify how it will be used during the user interaction. The dialogue description essentially includes two grammars: for recognition of an utterance on a specific domain, and for confirmation and possibly correction of a previously uttered phrase. Each grammar was updated during the acquisition phase to cover every user expression. Our approach for language modeling makes use of Recursive Transition Networks (RTN). These are finite state networks whose arcs allow linking other grammars in a recursive way. The resulting language is context free. Since the decoding step of an utterance can backtrack both the grammars and the words along the best path of

¹This work was performed while Marco Orlandi was visiting SRI Labs. Christopher Culy now works at FXPAL and his address is culy@fxpal.com.

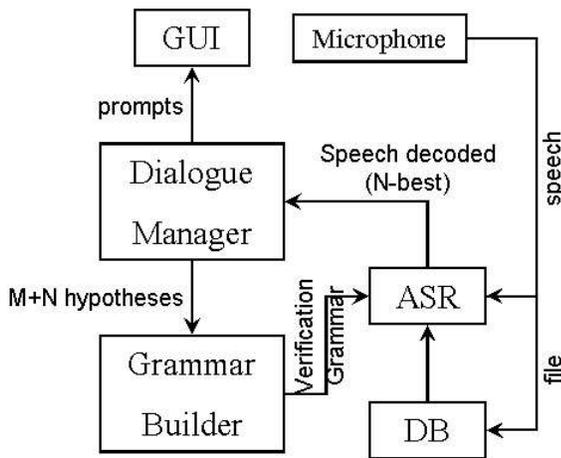


Figure 1: Dialogue System Architecture.

the language graph, the recognized string consists of a mix of words and structured information, i.e. it can be seen as a parse tree. The development of the understanding part of the system basically consists in designing a set of grammars.

A Graphic User Interface (GUI) is added for convenience, because any Text To Speech (TTS) engine can be included into the architecture and the user can read the system's prompts displayed.

Automatic Speech Recognizer (ASR) allows to perform speaker independent continuous speech recognition using RTN [5].

The Grammar Builder builds grammars in a proper format, each of them is loaded into the ASR.

The database (DB) stores speech waveforms, and in particular those related to ASR mistakes. Speech waveforms are used, in a separate phase, for recovering from errors.

The user interacts with the system using the GUI for reading prompts and a microphone for introducing inputs.

A dialogue domain, in a specific context, is formed by a set of simple concepts. For this work, we have defined and tested dialogue models for asking a specific context (e.g. address, date, etc.). During the interaction, the user has first to introduce all the mandatory concepts. Besides this, in a second turn, the user has to confirm to the system the correctness of the recognized utterance. The system can detect errors introduced by the recognizer because the user denies its hypothesis and is allowed to utter the correction. In the following we assume that when an error happens the recognized sentence is inside the N-best hypotheses (with N to be determined). This observation can be done for the confirmation phase as well. Using the above remark, when an error happens, we attempt to use the information that the recognized sentence was an error to remove that incorrect hypothesis and all the se-

mantic equivalent hypotheses from the allowable recognized strings. The Grammar Builder incorporates these constraints in a new recognition grammar using the remaining $(N - 1)$ hypotheses and the M hypotheses of the confirmation.

When an error is detected in a given turn (e.g. the user answers "no, ..."), a new grammar is built starting from the M hypotheses of the confirmation. This grammar allows to cover all the possible semantic units of the domain excluding, however, the one that was recognized to be wrong in the previous turn. This grammar is applied in a new rescoring phase where the waveform is provided by the acoustic database.

As an example a user's interaction in the "date domain" translated from Italian is reported below.

DIAL: Please, say a date.

USER: 22 december 2001

ASR :

-1- (DD(22)DD) (MM(december)MM) (YY(1 thousand (NUM3(1)NUM3))YY)

-2- (DD(22)DD) (MM(december)MM) (YY(3 thousands (NUM3(1)NUM3))YY)

-3- (DD(22)DD) (MM(december)MM) (YY(2 thousands (NUM3(1)NUM3))YY)

The system has asked the user for a date, who replied with December 22nd, 2001. The system has recognized December 22nd, 1001 and in the next step it asks for a confirmation. At this point the speech file has been stored in the database.

The dialogue continues in the following way:

DIAL: Do you want the date December 22nd, 1001?

USER: no 2001

ASR :

-1- (YN(no)YN) (YY(1 thousand (NUM3(1)NUM3))YY)

-2- (YN(no)YN) (YY((NUM3(21)NUM3))YY)

-3- (YN(no)YN) (YY(2 thousands (NUM3(1)NUM3))YY)

-4- (YN(no)YN) (YY(3 thousands (NUM3(1)NUM3))YY)

-5- (YN(no)YN) (YY(@a thousand (NUM3(1)NUM3))YY)

A standard dialogue system should prompt "Do you want the date December 22nd, 1001?" again, showing a rigid behaviour of the machine. But our system detects one or more errors introduced by the recognizer because the user denies the prompted hypothesis. Moreover the user has corrected the system, and the ASR produces $M = 5$ hypotheses of corrections. Two of the hypotheses (-1- and -5-) are semantically equivalent to the erroneous one, previously recognized. The Grammar Builder considers this observation and the denied sentence is removed from the new grammar, leaving: 22 december 21, 22 december 2001, 22 december 3002. The result of the new rescoring phase using the grammar above and the stored speech file

is:

ASR2: (DD(22)DD) (MM(december)MM) (YY(2 thousands (NUM3(1)NUM3))YY)

The final prompt is “Do you want the date December 22nd, 2001?”.

For testing the performance of the proposed method a measure called Accumulative Task Completion ($AccTC_i$) has been defined. It represents the portion of dialogues that succeed after n user utterances, where n ranges from 1 to a maximum number M of utterances. $AccTC$ can now be defined as follows. Firstly, utterances have been translated into semantic units and form the corpus update. They have been listed in a set S_i , where $i \in \{1, \dots, M\}$. The set S_i contains the semantic annotations of the utterances spoken after i sentences for each dialogue. In some ways S_{i-1} differs from S_i of the updates produced by the system after a confirmation phase. For each dialogue, a semantic annotation of the requests done by the user has been produced, this reference is maintained into a corpus. In a formal way $AccTC_i$ is defined as follows:

$$AccTC_i = 1 - \frac{SU_{iS} + SU_{iI} + SU_{iD}}{SU} \quad (1)$$

where SU is the total number of semantic units in the corpus annotation, and SU_{iS} , SU_{iI} , and SU_{iD} are the number of substitutions, insertions, and deletions that are necessary to make the (translated) update of the set S_i equivalent to (the translation of) the corpus update.

Given the domains “address” and “date”, if an error occurs, the system can manage more than one entry to correct its lack of understanding (i.e. the chosen domains contains more semantic units).

3. The Spoken Address Domain

The dialogue engine in the first prototype is a VoiceXML interpreter [3], developed in ITC-irst. The ASR is Dynaspeak™, developed in SRI [6]. This recognizer adopts JSGF (Java Speech Grammar Format) as formalism for grammars.

The system was totally developed in SRI Labs and deployed in the spoken address domain, in this case the task consists in understanding only a few concepts which set an address (street number, street name and city mandatory and zip code and state optional). In the past, SRI took a more standard approach to this task, with very good results [7]. We used a simpler, less accurate system for our data collection since we were concentrating on user corrections rather than address recognition per se.

At the beginning of the work we collected data and studied how users correct recognition errors during an interaction with the machine. We developed recognition grammars to match this behavior thus achieving a more natural way to interact in the presence of errors (see Fig. 2).

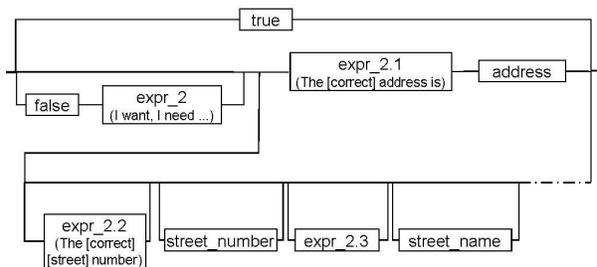


Figure 2: Description of the Confirmation Grammar.

The database used in the experiments is formed by 247 dialogues, 18 speakers, for a total of 760 speech files. 310 of these are correction sentences. These can be grouped according to three possible actions that the user can take when the system requires an explicit confirmation of the recognized hypothesis:

- a complete rejection (e.g. “no”) 6.13% of the sentences (19/310),
- a partial rejection, in other words new concepts are added to a rejection utterance (e.g. “no, it’s 333”) 53.87% (167/310),
- a new set of concepts (e.g. “333 Ravenswood Avenue”) 40.00% (124/310).

The grammars were updated after each user session. At the end of the data collection the performance increased from 64.37% to 72.87% on correctly recognized addresses.

Preliminary results show a reduction in number of dialogue turns to get a given error rate. After the first utterance, where a user speaks the address, the $AccTC_1$ is 73.65%; furthermore, if we consider the $AccTC_1$, for the first confirmation, it improves from 82.44% to 83.84%. If we consider the second confirmation phase $AccTC_2$ passes from 89.93% to 97.31%. Unfortunately, these two last results are not comparable because the data collected from the first dialogue system are not usable by the new one. This is due to different internal state of each dialogue system. But if we consider the subset of sessions where both the internal states of the two systems are aligned, the $AccTC_i$ passes from 95.21% to 99.26% and from 98.34% to 100% respectively for the first and second confirmation phases ($i \in \{1, 2\}$).

This preliminary work shows an improvement in the Accumulative Task Completion. It might be estimated between 7.97% and 84.55% on the first confirmation. Moreover it reduced the average number of dialogue turns.

# turns	# SU_c in $S_{noEC}^{beginner}$ (%)	# dialogues in $S_{noEC}^{beginner}$	# SU_c in $S_{EC}^{beginner}$ (%)	# dialogues in $S_{EC}^{beginner}$
0	271 (81%)	111	267 (80%)	111
1	101 (30%)	46	86 (25%)	38
2	37 (11%)	22	31 (9%)	18
3	27 (8%)	15	10 (3%)	10
4	18 (5%)	10	15 (4%)	10
5	11 (3%)	6	15 (4%)	9
6	8 (2%)	4	8 (2%)	7
7	6 (1%)	3	7 (2%)	7
8	0 (0%)	1	11 (3%)	6
9			2 (0%)	1
10			2 (0%)	1
11			1 (0%)	1
12			2 (0%)	1
13			0 (0%)	1

Table 1: Dialogues of “Beginners”.

# turns	# SU_c in $S_{noEC}^{experts}$ (%)	# dialogues in $S_{noEC}^{experts}$	# SU_c in $S_{EC}^{experts}$ (%)	# dialogues in $S_{EC}^{experts}$
0	284 (85%)	111	295 (85%)	115
1	71 (21%)	34	90 (26%)	39
2	30 (9%)	19	26 (7%)	14
3	23 (6%)	12	13 (3%)	9
4	12 (3%)	8	9 (2%)	6
5	9 (2%)	5	5 (1%)	4
6	5 (1%)	3	2 (0%)	2
7	3 (0%)	2	5 (1%)	2
8	5 (1%)	2	0 (0%)	0
9	2 (0%)	1		
10	0 (0%)	0		

Table 2: Dialogues of “Experts”.

4. The Spoken Date Domain

For a better evaluation of the ideas described above, we decided to do another data collection using a new dialogue system. Hence a new prototype was developed in ITC-irst Labs, able to recognize dates in Italian. This system uses a speaker independent continuous speech recognizer also developed in ITC-irst Labs. For this task the dialogue manager handles few concepts (a daytime, a month and a year) freely uttered.

The new prototype system was used to collect a database of dialogues uttered from a set of 14 speakers. Each speaker tried to set a date into the system. Each user was given a predefined set of dates and two sessions were collected using the machine. After the first session (*beginner*) the error correction capability was disabled or enabled, depending on which system was used first, and a new session was redone (*expert*). In this way dialogue turns with error correction are listed in $S_{EC}^{beginner}$ and S_{EC}^{expert} , whereas dialogue turn without error correction feature are listed in $S_{noEC}^{beginner}$ and S_{noEC}^{expert} .

The total number of dialogues in the database is 448, the number of speakers is 14 and on average each speaker has uttered 16 dates into both systems. Each dialogue includes three semantics units at maximum. Complete logs of all interactions (grammars used, system prompts, recognizer output, etc.) have been stored. For this task the ASR’s N-best feature was settled at 16. The length of the dialogues, expressed as number of turns, for each set is reported in Tables 1 and 2.

The Tables show, for each turn, the number of semantic units correctly recognized (SU_c) and the corresponding percentages. Besides, # dialogues indicates the number of active dialogues for each turn.

Results can be summarized as follows:

- the system with error correction used by a beginner speaker ($S_{EC}^{beginner}$), demonstrates a performance improvement. In fact the number of both semantic units correctly recognized after the first turn and dialogues successfully ended per turn are increased with respect to not using error correction

$(S_{noEC}^{beginner})$.

- By looking Table 2, an expert interaction exhibits a small decrease of the overall performance after the first couple of turns and a reduction of number of interactions between the machine and the user.
- Finally, note that $S_{EC}^{beginner}$ and S_{noEC}^{expert} contain the same speakers and dates uttered as $S_{noEC}^{beginner}$ and S_{EC}^{expert} . In these sets we can observe a reduction in the number of dialogues per turn.

Looking at this performance it seems fundamental having a good recording of the first turn, because the procedure of error correction uses it for recovering errors. During the data collection it happened that a speech waveform was badly recorded (i.e. for the presence of noise on the background, or for a mistake of the end-pointing algorithm). This file has decreased the dialogue performance on that sequence of turns. For this reason a mixed approach, based on a different strategy, should be introduced.

The next step is the manual transcription of the collected database and the process of $AccTC_i$, for each i .

5. Conclusions

Preliminary experimental results using this method show a reduction in the number of dialogue turns to get a given error rate.

The dialogue system can detect recognition errors during the confirmation phase after the first turn. It can adapt the language model to avoid errors and respond accordingly to the speaker.

6. References

- [1] M. Swerts, J. Hirschberg, D. Litman, "Corrections In Spoken Dialogue Systems". In *Proc. of ICSLP 2000*, Beijing, China, 16-20 October 2000.
- [2] D. Falavigna, R. Gretter and M. Orlandi, "A Mixed Language Model for a Dialogue System over the Telephone". In *Proc. of ICSLP 2000*, Beijing, China, 16-20 October 2000.
- [3] <<http://www.w3.org/TR/voicexml/>>.
- [4] J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, D. Byrd, "Analysis of User Behaviour under Error Conditions in Spoken Dialogs". In *Proc. of ICSLP 2002*, Denver - Colorado, USA, September 16-20, 2002.
- [5] F. Brugnara, M. Federico, "Dynamic Language Models for Interactive Speech Analysis". In *Proc. of Eurospeech 1997*, pp. 1827-1830, Rhodes, Greece, 1997.

- [6] Franco, H., Zheng, J., Butzberger, J., Cesari, F., Frandsen, M., Arnold, J., & Gadde, V. R. R., Stolcke, A., and Abrash, V., "DynaSpeak: SRI Scalable Speech Recognizer for Embedded and Mobile Systems". In *Proc. Human Language Technology Conference*, San Diego, CA, 2002.
- [7] Franco, H., Myers, G. Venkatraman, A., Frandsen M., "Spoken Address Recognition". In *SRI's Internal Report*, 2001.