The Simulation of a Human Operator by an ASP system

L.J.M. Rothkrantz R.J. van Vark
J.P.M. de Vreught S. Kruger

Knowledge Based Systems
Computer Science
Delft University of Technology
alparon@kgs.twi.tudelft.nl

Keywords: Dialogue Management, Simulation, Speech Recognition.

Abstract

This paper describes an application of Automated Speech Processing in two food ordering environments: drive–in restaurants and ordering by telephone. After describing a general model for such dialogues, the two environments are described in detail: covering word use and dialogue structure. Two prototypes have been developed for these environments. Testing these prototypes revealed that additional training of the speech recogniser is essential is this domain. Finally, a simulation study was done in which the human operator was replaced by a simulated system making use of a minimal set of responses selected from the recorded corpus. The simulation study showed that more or less standard dialogues can be simulated with such a system.

1 Introduction

For years Automated Speech Processing (ASP) was the exclusive domain of researchers in research labs and universities. Recently ASP has been successfully applied in commercial applications such as telephonic inquiry systems, dictation systems, and call centres. In these applications, the human voice can be used as an interface to send commands to a computer, to dial the phone, or operate the radio in a car. In all those applications the human–machine interaction has a simple structure. Simulating human–human dialogues in all its complexity is far from being realised.

In this study ASP is applied in a restricted context. Most fast–food restaurants have the facility to order meals from a special booth or by telephone. Currently, these command desks are run by a human operator. The goal of this study is to simulate the human operator by using an ASP system.

Our approach will be corpus–based, i.e. we take the current customer–human operator dialogues as our starting point and try to mimic the behaviour of the human operator by the ASP system. However, regarding the state of the art in speech technology, some simplifications have to be made. The current human–human dialogues have evolved to a very effective and efficient system with a high user appreciation rate. By simulating these human–human dialogues, the ASP system is also expected to be highly appreciated by the customers.

2 Dialogue Model

The first environment examined for application of ASP, is ordering food in drive–in restaurants. Human–human interaction appeared to be structured quite simple as customers are faced with a billboard or card showing the menu choices. When the customer is asked to order, he uses short commands following the information on the billboard or menu card. After ordering, the operator will verify the order or provide information on prices or the food retrieval desk.

The second environment is a meal ordering system by telephone. As no menu information is available to the customer, the dialogues are more complex than in the first environment. Instead of ordering meals by their official name, customers give a description of

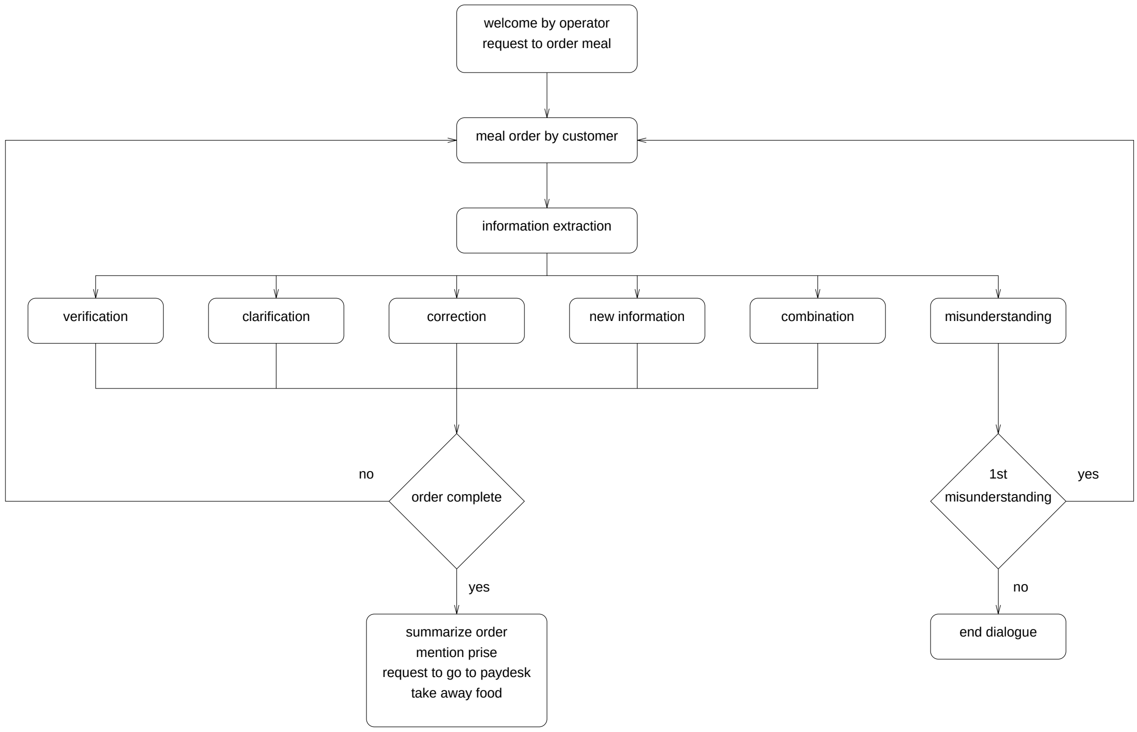

Figure 1: Data flow in a general dialogue

the meal or a list of ingredients. For example, ‘Pizza x with salami’ or ‘Pizza x, is that with salami?’ are completely different in meaning, but use the same keywords. These cases can be distinguished using prosodic information which is a hot topic in speech research.

The operator’s task is goal–directed as he has to fill out an order form representing the requested meals and drinks. The start of a dialogue is non–directive and the initiative is left to the customer. Keywords are selected from the customer’s utterances. These keywords include names of meals and ingredients, as well as general keywords like numbers. If the customer’s order needs to be disambiguated, the operator phrases additional questions and takes the initiative. In figure 1 the data flow of a general dialogue is displayed. The operator or ASP system can undertake several actions:

Verification The information extracted from the customer’s utterances is always verified. For example, ‘You want two pizzas x?’

Clarification Ambiguity is reduced by requesting additional information or verifying the given interpretation. For example, ‘Do you want one or two pizzas x?’

Correction When an unavailable menu option is ordered, a correction is requested. For example, ‘Menu x cannot be combined with French fries, do you want them separately?’

New information More orders will be requested until the customer indicates that he wants to stop. For example, ‘Do you also want something to drink?’

Combination Prompts of different categories will be combined when possible. For example, to combine a verification (menu x) with a next question (to order French fries): ‘Do you want your menu x with French fries?’

Misunderstanding When no information can be extracted from the last utterance, the customer will be asked to repeat the information. If misunderstanding occurs repeatedly, the dialogue will be stopped.

The vocabulary used in the interaction is limited. In the first environment, the vocabulary consists of approximately 100 words. In the second environment, most dialogues can be handled using a vocabulary of approximately 400 words.

When the operator takes the initiative during the dialogue, the customer is supposed to answer questions. It cannot be expected that the utterances of the customer are grammatically correct sentences. The commands are few words statements and the utterances of the customer will have all the characteristics of spoken language which is quite different from written language.

3 Automated Speech Processing

Developing an ASP system is very complicated and time–consuming. ASP development consists of a number of steps. The first step is to analyse the environment and domain in which the ASP system is going to function. This analysis is done by developing a model, a grammar, and a vocabulary. Especially the latter two are complicated as they are usually developed using a corpus–based approach, which involves recording a large number of dialogues in a similar environment. The second step is to develop the different components of the ASP system (see figure 2). These components are all tailored to the specific domain of the ASP system. The final step is to train certain components of the ASP system using data collected in the first step. An example is training a speech recogniser using the defined vocabulary.

3.1 ASP environments

An ASP system was developed for two different environments. In two restaurants, a total of 400 dialogues between a human operator and customers have been recorded. Half of the dialogues cover a speech booth environment at a drive–in restaurant (Env_1) and half the dialogues cover a telephone order system (Env_2). The average length of a dialogue is 87 words, of which 55 are uttered by the customer and 32 by the operator.

The transcribed text of these dialogues has been analysed in two different ways. In a first approach, keywords are extracted from every customer's utterance: names of the possible meals, ingredients, and toppings, and specific words such as numbers and 'with' or 'without'. In Env_2, specific commands are addressed by additional options like time expressions, for instance 'as soon as possible' and 'about 6 o'clock' (see table 1).

	names menus	ingredients add-ons	additional words
Env_1	23	17	52
Env_2	61	32	416

Table 1: Average number of customer words

From the recorded dialogues bigram and trigram models were generated, as well as grammatical structures of sentences. If the customer uses one of the standard dialogues, the meal order can be generated from the extracted keywords. In Env_2 more than half of the recorded dialogues are non–standard. Keyword extraction does not provide correct or sufficient information, for instance when discussing prices, ways of paying, special wishes, and during social talk.

By restricting dialogues to the standard script some of the non–standard dialogues can be converted to standard dialogues.

In the non–standard dialogues more information has to be extracted from the dialogues and for a successful interpretation of these (partial) information a knowledge based system is needed with a reasoning module. To be able to reason with such a system, the dialogue utterances have to be coded. A similar coding scheme was developed in the design of an automated system for providing information about public transport [5].

3.2 L&H–prototype

The first prototype is based on Lernout & Hauspie speech recognition software [1], which is a multi–lingual platform using a 'recogniser engine' based on phoneme recognition. Using the L&H's Application Programmer's Interface, an application can be implemented easily. At run–time only a part of the vocabulary and grammar will be activated. Unfortunately there was no module for dialogue management available based on the blackboard techniques we developed for our own dialogue manager [6], which general design is depicted in figure 2.

3.3 Disciple–prototype

Disciple was developed by one of the authors within the framework of a speech recognition project at Rutgers University [3]. The module for the English language is trained on the Timit database [2].

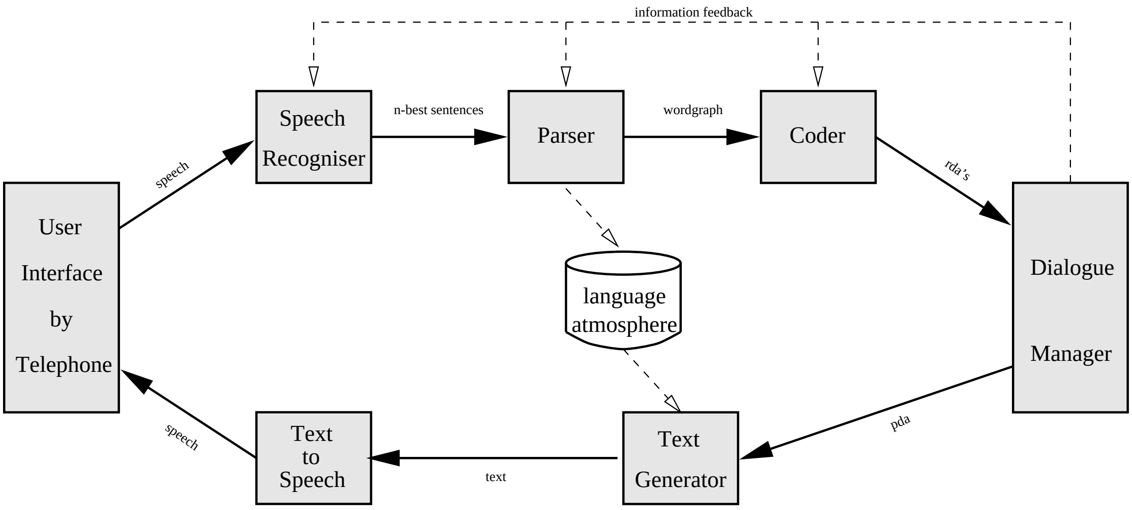

Figure 2: Model of ASP system

For the Dutch language Delft University of Technology used public databases and proprietary ones. For optimal performance additional training is necessary. For DISCIPLE the sources were available, so embedded training can be applied, in contrast to the Lernout & Hauspie software.

3.4 Test result

50 students were selected from the Delft University of Technology student population to test the performance of the developed prototypes. The students were asked to read aloud 50 words (names of menus and ingredients) and 50 sentences. The recognition results for both the L&H–prototype and DISCIPLE–prototype are shown in table 2.

	speech recognition rates		
	words		sentences
	untrained	trained	untrained
L&H	29 %	–	72 %
DISCIPLE	32 %	85 %	–

Table 2: Recognition rates in Env_2

Table 2 shows that additional training is necessary to achieve a high recognition rate. The names of menus are mostly Italian words, which are difficult to recognise using a speech recogniser trained for common English.

4 Simulation Experiment

The goal of our study was to simulate a human operator in a meal ordering application. Therefore, an ASP system has to be developed connected to the public telephone network.

From the recorded dialogues all the utterances of the human operator are selected. This set of utterances is reduced to a minimal set by skipping rarely used utterances. The minimal set of operator utterances is used to select an answer to every customer utterance. Selecting the appropriate utterance is done by the dialogue management module.

The question is whether we can simulate a human operator with this minimal set of utterances. This can be tested in a so called Wizard of Oz experiment [4]. In such an experiment, the ASP system is secretly operated by a human. All responses are activated by pressing buttons, which are grouped together in a graphical user interface.

In an experiment, 25 respondents were asked to order a meal using an ASP system. They were requested to call the service twice. During the first call, non–verbal pictures were displayed to show which meals they had to order. During the second call, they were free to choose their own meal. It was proven that a meal was successfully ordered as long as the dialogue fits the standard pattern.

In the non–standard situation when the customer starts up a highly interactive dialogue the operator runs out of preselected utterances. The dialogue has to be restricted to the script 'how to order a meal'. Evaluation showed that all respondents had the feeling of talking to a computer. Afterwards they were not informed of the fact that the system was simulated by a human.

5 Conclusions

It was shown that it is possible to simulate a human operator in a meal ordering environment by an ASP system as long as the dialogues are more or less straightforward. In a complex situation customers do not know how to order meals or have no idea what meals are available. For these situations the system fails.

At this moment there is a project running at the Delft University of Technology to develop a dialogue manager to handle complex, mixed–initiative dialogues. The keyword extracting approach is too simple to provide the dialogue manager with sufficient information. The utterances have to be abstracted to linguistic classes. An information based coding scheme was developed for transport information dialogues. The coding scheme has to be adapted to the current application. For a high recognition rate, additional training of speech recognisers is absolutely necessary.

References

[1] D. van Compernolle, *Speech Recognition at Lernout & Hauspie,A technological perspective*, Technical Report L&H–SR–PI–96–001, Lernout & Hauspie, 1996.

[2] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, N.L. Dahlgren, *Darpa Timit, Acoustic–Phonetic Continuous Speech Corpus, NIST Speech Disc 1-1.1*, U.S.. Department of Commerce, 1993.

[3] S. Kruger, DISCIPLE*, a command system based on ASR*, Master's thesis, Delft University of Technology, to appear 1997.

[4] L.J.M. Rothkrantz, W.A.Th. Manintveld, M.M.M. Rats, R.J. van Vark, J.P.M. de Vreught, H. Koppelaar, *An Appreciation Study of an ASR Inquiry System*, To appear in the Proceedings of the EuroSpeech 1997 Conference, 1977.

[5] R.J. van Vark, J.P.M. de Vreught, L.J.M. Rothkrantz, *Classification of Public Transport Information Dialogues using an Information Based Coding Scheme*, Workshop Dialogue Processing in Spoken Language Systems, European Conference on Artificial Intelligence, 92–99, 1996.

[6] R.J. van Vark, J.P.M. de Vreught, L.J.M. Rothkrantz, *An Automated Speech Processing System for Public Transport Information Services*, Third International Congress on Information Engineering, 212-221, 1997.