Simulation and Recognition of Information Dialogues using Markov Models*

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

Delft University of Technology
Faculty of Technical Mathematics and Informatics
Julianalaan 132, 2628 BL Delft, The Netherlands
e-mail: alparon@kgs.twi.tudelft.nl

ABSTRACT

OVR, a Dutch provider of information concering national public transport, has started an investigation of the usability of Automated Speech Processing to automate part of their services in order to be able to serve all customers satisfactory. Meanwhile, a corpus recorded during this research is used to develop a training simulator for training their human operators to handle conversations more efficient. Based on the corpus a coding scheme was developed to analyse public transportation information dialogues. To gain better insight in how dialogues are taking place, Markov models are developed using this coding scheme. These models are used to cluster dialogues and to develop scripts which ideal dialogues must follow. The Markov models derived to be used in the training simulator can later be used in the dialogue management component of the ASP system.

INTRODUCTION

Last year OVR, a Dutch provider of information concerning national public transport, started an investigation of the usability of Automated Speech Processing (ASP) in their company. As a result of the increasing load at call centers, operators have to handle conversations more and more effectively. Therefore in the future part of their humanly operated information service would have to be automated by using ASP. Meanwhile we also like to develop a simulator for training operators. By using such a simulator operators are trained to minimize call duration and maximize client satisfaction by handling conversations more efficiently and effectively.

The simulator will be based on the analysis of a large corpus of over 5000 dialogues recorded at call centers of OVR. Using this large corpus a Markov model is designed to model information dialogues. Such an Markov model serves a number of purposes:

- clustering of dialogues,
- developing control mechanisms to identify 'valid' dialogues,
- developing scripts which ideal information dialogues must follow.

For the future ASP system the Markov models give us insight into the probabilities of dialogue parts.

Related work is discussed first, followed by a description of the dialogues examined and the coding scheme used to analyse these dialogues. In the fourth section, the Markov model based on these dialogues will be presented. In the fifth section we will discuss results of applying this Markov model to our coded corpus. Finally, future work will be described.

RELATED WORK

Carletta et al. [CII+95] developed a coding scheme for task oriented dialogues existing of three levels. At the highest level dialogues are divided in transactions or subdialogues corresponding to major steps in the participants' plan for completing the task. At the intermediate level conversational games consist of initiations and responses. A conversational game is in fact a sequence of utterances, starting with an initiation and ending with the requested information being transferred. At the lowest level conversational moves are utterances which are initiations and responses named according to their purpose. Our coding scheme has been strongly influenced by their scheme. However, our scheme is both more generic and more detailed.

In the VERBMOBIL project researchers developed a taxonomy of dialogue acts [AR95]. To model the task oriented dialogues in a large corpus they assumed that such dialogues can be modelled by means of a

*This research was funded by Openbaar Vervoer Reisinformatie (OVR) and SENTER.

limited but open set of dialogue acts. They defined 17 dialogue acts and semi-formal rules to assign them to utterances. These assignment rules served as a starting point for the automatic determination of dialogue acts within the semantic evaluation system. From the analysis of an annotated corpus of 200 dialogues they derived a standard model of admissible dialogue act sequences. We also investigated phase sequences in a similar way.

Based on the herefore mentioned papers we have developed a new coding scheme [vVdVR96]. The primary goal of that coding scheme was to analyse existing OVR dialogues [RVV+96]. Results of this analysis enable us to develop a speech recognition system and in the meantime these results will also be used to build a training simulator. To enhance the performance of such systems, a better insight into dialogue probability is needed.

DIALOGUES

The coding scheme for training the Markov model was developed based on a set containing 5205 dialogues [RVV+96]. These dialogues consist of telephone conversations between two human agents. One of these agents is an operator working at a call center, while the other agent is a client calling the call center to gain information concerning public transport services.

Goals of the current research project are to identify commonly used dialogue structures based on statistical relationships between phase and move transitions. These relations are used to train an ASP system while the dialogue structures will be used to cluster dialogues and develop scripts to build a simulator for training operators.

OUTLINE

A coding scheme was developed to identify the underlying dialogue structures. The scheme is based on the dialogue acts used to transfer information between agents. A dialogue act is based on a grammatical sentence. Several of these dialogue acts, or moves, make up the utterances of each of the agents. Also, moves have several arguments like information codes and markers for interruptions [vVdVR96]. However, these arguments are not relevant for the Markov model presented later.

PHASES

Dialogues can be divided in several phases or subdialogues which accomplish one step in retrieving the desired information. Phases are important aspects of our coding. A dialogue phase is a good indication of the flow of information at a certain time in the dialogue. For instance, during the Greeting phase both agents give information, while in the Query phase most information is given by the client. In our coding scheme, six phases can be distinguished:

- **Greeting (G)** This is the first phase of all dialogues in which both parties exchange greetings.
- **Query (Q)** In the Query phase the client has the initiative and tries to pose his question.
- **Pause (P)** The Pause phase indicates the operator is searching. This phase can often be found between Query and Information phase.
- **Information (I)** Information is provided to the client during this phase and therefore the operator has the initiative.
- **Subquery (S)** During this phase only a small piece of the previous Query is varied. The Subquery phase helps to keep track of incomplete queries.
- **Goodbye (B)** This phase marks the end of a dialogue and consists of exchanging thanks and saying goodbye.

MOVES

Phases consist of one or more conversational games [CII+95]. These games usually consist of initiations, e.g. questions or statements, followed by responses, like answers or acknowledgements. Some moves of our coding scheme are discussed below as they are important to define inherent dialogue structure.

- **Greeting (Gre)** Greeting is the move in which an agent greets the other agent and often introduces himself.
- **Question (Que)** Whenever an agent asks for a piece of information, this is coded as a question. Information can be introduced as part of a question.
- **Statement (Sta)** Statements are utterances with the only purpose of presenting information, often in response to a question posed earlier in the dialogue.
- **Check (Che)** A check requests an acknowledgement of information the checker has some reason to believe.
- **Alignment (Ali)** An alignment verifies the attention and/or agreement of the partner to move to a next phase.

- **Bye (Bye)** The bye move describes the social ritual where the client thanks the operator, who on her term expresses that she was glad to help, after which both parties say goodbye.

Other dialogue moves are Acknowledgement, Clarification, Reconfirmation, Pause, and Other.

MARKOV MODEL

The coding scheme was originally applied to a sample of 483 dialogues. We have chosen not to code 37 of them since they would not help us designing a Markov model of information dialogues. Although in theory we could have coded these dialogues, we have chosen to call these 37 dialogues 'uncodable.' The remaining dialogues were used for this paper.

MODEL

Our approach to build a Markov model will be demonstrated using a general symbool m_i to describe the i-th move in the dialogue. Later, this general symbol can be replaced by specific move descriptors we would like to use. A move can be described using the phase in which the move took place, the dialogue act, and which agent produced the move. Of course, combinations of these move descriptors can also be applied. The main difference between move descriptors is complexity: a 7 state Markov model with 23 edges for phases alone versus a 48 state Markov model with 477 edges for a combination of phases and moves.

We adopted a technique often used in speech recognition. A speech recognition system tries to find words such that the probability of those words given the speech signal is maximal:

$$\Pr(words|signal) = \frac{\Pr(words)\Pr(signal|words)}{\Pr(signal)}$$

The term $\Pr(signal)$ is a constant and can be disregarded. The parts $\Pr(words)$ and $\Pr(signal|words)$ are resp. called the language model and the acoustic model.

In our case, we would like to determine the most probable move descriptor given a certain dialogue.

$$\Pr(m_i's|dialogue) = \frac{\Pr(m_i's)\Pr(dialogue|m_i's)}{\Pr(dialogue)}$$

In this paper we will concentrate on $\Pr(moves)$, leaving the 'acoustic' counterpart for further research. The term $\Pr(moves)$ can also be written as:

$$\Pr(m_1 \dots m_n) = \prod_{i=1}^{n} \Pr(m_i|m_1 \dots m_{i-1})$$

This last terms is pretty awkward to handle, but there are some popular simplifications possible that seem to work in practice. One effective approach is the bigram model where

$$\Pr(m_i|m_1 \dots m_{i-1}) \approx \Pr(m_i|m_{i-1})$$

So after simplification we have:

$$\Pr(moves) \approx \prod_{i=1}^{n} \Pr(m_i|m_{i-1})$$

Like in speech recognition, we will also examine the usability of trigrams or n-grams in general. The Markov model is not solely dependant on the current state but on the last $n-1$ states. In formula:

$$\Pr(m_i|m_1 \dots m_{i-1}) \approx \Pr(m_i|m_{i-n+1} \dots m_{i-1})$$

When we write $m_{i-n+1} \dots m_{i-1}$ as $df_{i,n-1}$, this can be written as:

$$\Pr(m_i|m_1 \dots m_{i-1}) \approx \Pr(m_i|df_{i,n-1})$$

Although n-grams seem to violate the Markov property, a finite history of $n-1$ move descriptors can be coded into the states, and thus retaining the Markov property.

CALCULATING ACCURACY

To assess the model's value methods are needed to calculate its accuracy. Therefore, we have developed a method to derive this accuracy using the accuracy to predict the next move descriptor given the $n-1$ preceeding move descriptors. We introduce the term tag t_i which is the move descriptor we predict its tag, given $n-1$ preceeding move descriptors. This prediction is accurate if the observed move descriptor m_i is equal to the tagged move descriptor t_i. To calculate the probability of a correct prediction (c.p.) we use:

$$\Pr(c.p.|df_{i,n-1}) = \Pr(m_i = t_i|df_{i,n-1})$$

However, this is just the accuracy of a single prediction instead of the accuracy of the model. To obtain the accuracy of the model, we have to add all chances of correct prediction given a certain dialogue, weighted by the chances of that dialogue occurring:

$$accuracy = \sum_{\forall df_{i,n-1}} \Pr(c.p.|df_{i,n-1}) \cdot \Pr(df_{i,n-1})$$

The accuracy computation will result in a percentage of correct predictions of the next move descriptor. Although we investigated n-grams of different size seperately, one really would like to obtain a model in which predictions based on different sized n-grams can be combined to increase accuracy. One would like to weigh different predictions, resulting in ($\Sigma\lambda_i = 1$):

$$\Pr(t_i|df_{i,n-1}) = \sum_{j=1}^{n} \lambda_j \Pr(m_i|df_{i,j-1})$$

Charniak [Cha93] described methods to estimate these factors λ_j using hidden Markov models. Extending our Markov model with combinations of different sized n-grams will be a topic for future research.

RESULTS

In this section results will be presented of applying the above theory to our corpus of coded dialogues. Three types of move descriptors will be used: phase, phase combined with dialogue acts, and phases, dialogue acts and agents. Accuracy measures will be calculated for 1-grams, 2-grams, and 3-grams, while the text only discusses bigram models of the different types of move descriptors.

PHASES

The first step we made was to count the number of phase transitions that actually occur in the dialogues (see table 1). Furthermore, it is known that every dialogue starts with a G phase and ends with a B phase.

Table 1: Phase transitions

	G	Q	S	P	I	B	total
G	619	440	0	0	0	0	1059
Q	0	2461	275	418	317	5	3476
S	0	306	285	62	269	0	922
P	0	153	73	422	397	0	1045
I	0	107	280	143	5330	491	6351
B	0	9	9	0	38	2053	2109

Normalising all values using row totals, the table results in the estimated probabilities used in the bigram model: $\Pr(m_i|m_{i-1})$. As all dialogues start with a G phase, the chances of predicting the first phase correctly are 1. The final state in the Markov model is characterised by the fact that all incoming edges are labelled B, because all dialogues end with a B phase. A graphical representation of the corresponding Markov model can be found in figure 1.

PHASES AND MOVES

Combining phase and dialogue acts to derive move descriptors resulted in the figures presented in the table on the next page. It is known that every dialogue starts with a Greeting move in the Greeting phase, while all dialogues end with a move descriptor at least consisting of the Bye phase.

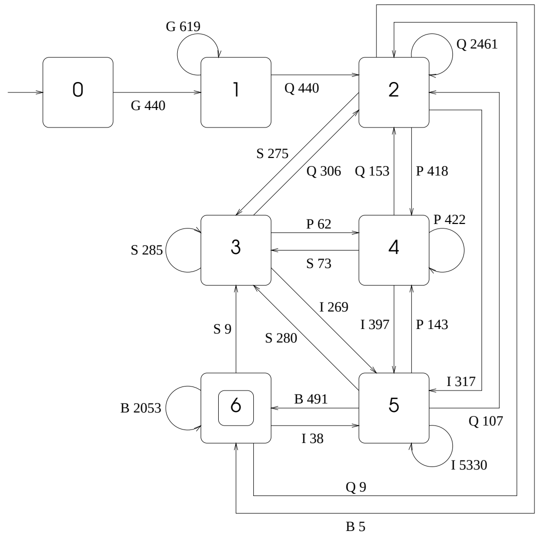

Figure 1: Phase Markov model (initial state = 0, final state = 6 and transitions are labeled by phase and the number of transitions found in the sample.)

The automaton related to this Markov model is too complex to be shown in a figure. However, looking at the table one can see localised groups or clusters of move descriptors. The most spectacular clusters correspond to the Query, Information, and Bye phase. Usually, the dialogue "moves" within such a cluster for a number of moves. When the dialogue proceeds to the next cluster, it usually does not return to a previously visited cluster, unless totally new questions are asked.

ACCURACY

Using the formulaes presented above, accuracy measures can be derived for the three models. These measures are all taken using both a 1-gram model (i.e. using no dialogue fragments) as well as 2-gram and 3-gram models.

Table 2: Prediction accuracy (Ph = phase, Da = dialogue act, Ag = agent)

	1	2	3
Ph	40 %	73 %	76 %
Ph + Da	18 %	48 %	52 %
Ph + Da + Ag	13 %	48 %	52 %

A first glance at table 2 gives the impression that the used methods result in poor accuracy. Only applying the Markov model based on phases results in satisfactory predictions. However, predicting the next phase from a certain move descriptor is much easier as predicting the next dialogue act. An example is the descriptor Q-Sta, i.e. a statement in the Query phase. In this case 791 of the next 1011 moves return in the Query phase, resulting in 78 % accuracy. Predicting the next dialogue act results in a correct prediction in only 416 of the total cases, i.e. in 41 % accuracy. Therefore, we might examine predicting less information using more complex move descriptors.

Another promising technique might be to predict not just 1 move descriptor but n move descriptors with a high enough probability. This technique might be described as n-best tagging. We introduce a tag set, T_i, which contains all n-best tags for the next move descriptor. When we look at k-grams, the probability of a correct prediction can now be written as:

$$\Pr(c.p.|d_{i,k-i}) = \Pr(m_i \in T_i|d_{i,k-1})$$

Using 3-best tagging for the move descriptor Q-Sta will result in a prediction accuracy of 64 %. Another clear example of the benefits of this strategy is predicting using the move descriptor G-Gre. Using only 1 prediction the accuracy is only 33 %, while using 3-best tagging results in 79 %. In the case of 4-best tagging, the result would even be 97 %. Clearly, the technique of n-best tagging has to be examined in more detail. Possibly, not a fixed number of best tags can be used, but the predictions can be picked based on individual accuracy. For instance, all predictions with an individual accuracy of over 20 % could be used.

FUTURE RESEARCH

One of the goals in the Alparon project is to develop a simulator which will be used for training novice operators, supervising operators, and for assessment of efficiency and effectiveness of operators. The second goal is to develop an ASP system for handling public transport information dialogues. To achieve these goals, several subgoals were planned along the way, of which the first was the development of the previously described Markov models. However, to achieve satisfactory results we will have to integrate Charniak's ideas into our model using a Hidden Markov model to derive weighing factors. We will also have to investigate the use of n-best tagging.

The second step is the development of information dialogue scripts and collecting data about the probabilities of dialogue parts. These scripts will describe "valid" information dialogues. Valid information dialogues are dialogues which are handled effectively by the operator. These scripts can be used to create numerous dialogues suitable for training operators.

Thirdly, a training simulator will be developedi, which most importantly will handle the client side of the conversations. This system will create create dialogues based on scripts. Therefore, the system has to be able to react to the trainee's actions.

As a fourth step we like to develop an expert system based on the knowledge of experienced operators and we like to measure effectiveness and efficiency of the dialogues: flexibility and dialogue handling, duration of dialogues and expressions, and complexity and diversification of utterances. We will test this simulator and make some simulation experiments.

Finally, a speech recogniser will be built using the Markov models to control the dialogues. This recogniser can also be incorperated into the simulator, making its interface to the operators more realistic.

References

[AR95] J. Alexandersson and N. Reithinger. Designing the dialogue component in a speech translation system, A corpus based approach. In *Corpus-based Approaches to Dialogue Modelling*, 9th Twente Workshop on Language Technology, pages 35–43. University of Twente, 1995.

[Cha93] E. Charniak, editor. *Statistical Language Learning*. MIT Press, 1993.

[CII+95] J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson. The coding of dialogue structure in a corpus. In *Corpus-based Approaches to Dialogue Modelling*, 9th Twente Workshop on Language Technology, pages 25–34. University of Twente, 1995.

[RVV+96] L.J.M. Rothkrantz, R.J. van Vark, J.P.M. de Vreught, J.W.A. van Wees, and H. Koppelaar. Automatische spraakherkenning en OVR-dialogen. Technical report, Delft University of Technology, Knowledge Based Systems, 1996. in Dutch.

[vVdVR96] R.J. van Vark, J.P.M. de Vreught, and L.J.M. Rothkrantz. Classification of public transport information dialogues using an information based coding scheme. In *Workshop Dialogue Pro-*

cessing in Spoken Language Systems, European Conference on Artificial Intelligence 1996, 1996.