AUTOMATIC CLASSIFICATION OF DIALOGUE CONTEXTS FOR DIALOGUE PREDICTIONS

Cosmin Popovici*, *Paolo Baggia*$^{\bullet}$ *Pietro Laface*$^{\diamond}$ *and Loreta Moisa*$^{\diamond}$

$^{\bullet}$ CSELT - Centro Studi E Laboratori Telecomunicazioni
Via G. Reiss Romoli 274 - 10148 Torino, ITALY, e-mail: baggia@cselt.it
$^{\diamond}$ Dipartimento di Automatica e Informatica - Politecnico di Torino
Corso Duca degli Abruzzi 24 - 10129 Torino, ITALY, e-mail: laface@polito.it
* I.C.I. - Institutul de Cercetari in Informatica
Bd. M. Averescu, 8-10 - Bucuresti, ROMANIA

ABSTRACT

This paper exploits the broad concept of dialogue predictions by linking a point in a human-machine dialogue with a specific language model which is used during the recognition of the next user utterance. The idea is to cluster several dialogue contexts into a class and to create a specific language model for each class.

We present an automatic algorithm based on the minimal decrease of mutual information which clusters the dialogue contexts. Moreover the algorithm is able to guess an appropriate number of classes, that gives a good trade off between the mutual information and the amount of training data.

This automatic classification procedure allows the full automatic creation of context-dependent language models for a spoken dialogue system.

1. INTRODUCTION

The statistical language model (LM) our paper deals with, is designed for task-oriented Spoken Dialogue Systems (SDSs). The variety of the possible user answers to the system questions is limited because the interaction takes place in the fixed domain imposed by the application.

For the translation/dictation systems an usual technique is to use general training databases like WSJ-NAB, but in our case, such general data do not produce good results.

In [11] it is already proved that the use of task-dependent databases reduces of an order of magnitude the perplexity values, in comparison with the use of a general task-independent training corpus. Moreover, using the same task-dependent training corpus, the perplexity value can be further reduced, considering the specificity of each dialogue context (in [4] a decrease of about 30% is presented).

Our approach to create a dialogue-context dependent LM, is similar to the static prediction algorithm presented in [2]. Other works that take advantage from the dialogue context in the environment of SDS are [10], [6], [5].

If unconstrained human-human spontaneous speech conversations are analized, such as in [9], several models are proposed for the automatic detection of dialogue acts in conversational speech and several experiments are described for using the detector to switch among the 42 dialog-act specific trigram LMs. The results are encouraging but not statistically significant at least in the Switchboard corpus of conversational telephonic speech [8].

The concept of dialogue-dependent LM is based on the fact that many dialogue contexts produce similar user answers. So we can cluster these contexts into classes and create for each class a specific LM. So far, in our system the classification of dialogue contexts was made manually (see [12]). In this paper we present an algorithm for automatic classification.

Our algorithm performs a clustering procedure beginning with one class for each context. For this classification we obtain the maximal value for the mutual information between the set of events and the set of event classes. The merges are selected in order to produce the minimal decrease of mutual information. The algorithm is also able to guess an appropriate number of classes, that gives a good trade off between the mutual information and the amount of training data used for each specific LM.

Although the dialogue dependent LMs are used in the Swiss Time Table System on a graph of concepts and not on at word level, the clustering algorithm described in [5] is also based on the mutual information, but it deals with a fixed number of classes. In that case the elements are moved from one class to another, in order to maximize the mutual information.

In Section 2. we present an overview of Dialogos, a task-independent SDS, on which the task-dependent applications developed in CSELT are based. A description of the dialogue contexts will be illustrated on an excerpt from an interaction in the Italian Time Table Railway Inquiries System. Section 3. presents the clustering algorithm. Section 4. contains the comparison between manual and automatic classification.

2. SYSTEM OVERVIEW

Dialogos is a software only, completely integrated SDS which runs in real-time both on DEC-Alpha and PC-Pentium platforms equipped with a Dialogic D41E Board. A telephone interface connects the acoustical front-end and the synthesizer to the public telephone network, while the dialogue manager is connected with the application database.

The acoustical front-end performs feature extraction and acoustic-phonetic decoding. The recognition module is based on a frame-synchronous Viterbi decoding, where the acoustic matching is performed by a phonetic neural network [7].

At each dialogue step, based on the current dialogue context, the dialogue manager selects the corresponding Dialogue Prediction that activates the specific LM from a set of statistic trigram class-based LMs. The selected LM uses only the bigram probabilities during the recognition phase, while its trigrams probabilities are applied for rescoring the N-best hypotheses.

The linguistic processor starts from the best-decoded sequence and performs a multistep robust partial parsing [3]. At the end of the analysis, it builds the deep semantic representation of the user utterance, that is sent to the dialogue manager [1]. The goal of the dialogue manager is to obtain the instantiation of the parameters needed for the database inquiry. So, it interprets the deep semantic representation, and, based on the dialogue history and on the active focus, it chooses a new dialogue context. According to this information (dialogue context), the dialogue manager generates also the system sentence, to be synthesized.

A dialogue context is determined by the dialogue act (DA) (such as REQUEST and CONFIRMATION) applied to a set of parameters (see Figure 1). In our system these information are explicitly represented so they can be directly used to constraint the recognition of the next utterance. While in the recognition of human-human conversations, the DA of an utterance has to be either automatically or manually detected. The classification proposed in [9] is based on 42 DAs types (such as statement, question, answer, backchannel, agreement, disagreement, etc), which are domain independent. In our case the DAs are only 3 (REQUEST, CONFIRMATION and REQUEST-AND-CONFIRMATION) and they are used in conjunction with domain parameters, so that our classification is mainly a domain dependent one.

Combining the 8 parameters of the Train Time Table Scheduling application (e.g: departure_city, arrival_city, relative_day, week_day, day_and_month, hour, part_of_day, time_interval) we can obtain more than 160 possible dialogue contexts, but our system strategy handles only 70 of them. Even in a large training database, many of these dialogue contexts have a small number of instances. A good approach to solve this problem is given by the clustering of the dialogue contexts [12].

In the off line training phase, each dialogue context is assigned to a Dialogue Context Class (manually or using the here presented classification algorithm). The creation of the set of specific LMs is based on a database containing the user utterances tagged with the dialogue contexts in which they were given.

The database utterances which are tagged with dialogue contexts that belongs to the same dialogue context class are selected to train a specific LM.

At run time a specific LM is activated by a dialogue prediction (the index of the dialogue context class) sent by the dialogue manager to the recognition module. The recognizer switches to the corresponding specific LM.

S1: Hello, This is Railway Timetable Enquiry Service.
Where are you leaving from and going to ?
Dialogue_Context: REQUEST: departure_city, arrival_city
Dialogue_Context_Class: Class_3
U1: From Turin to Milan on Sunday morning.
Recognized Sentence: From Turin to Milan tomorrow morning.
S2: Are you leaving from Turin and going to
Milan tomorrow morning ?
Dialogue Context: CONFIRMATION: departure_city, arrival_city, relative_day, part_of_day
Dialogue Context Class: Class_5
U2: No, I am leaving on Sunday.
Recognized Sentence: No, I am leaving on Sunday.
S3: Are you leaving from Turin and going to
Milan on Sunday morning?
Dialogue Context: CONFIRMATION: departure_city, arrival_city, week_day, part_of_day
Dialogue Context Class: Class_5
U3: Yes.
Recognized Sentence: Yes.
S3: I have found one connection ... Do you want other
information about this connection
Dialogue Context: CONFIRMATION: connection_information
Dialogue Context Class: Class_7
U4: No, thanks.
Recognized Sentence: No, thanks.
S5: Thank you for the call. Good bye!

Figure 1: Excerpt from a dialogue

3. AUTOMATIC CLASSIFICATION ALGORITHM

In this section we present a automatic algorithm for clustering a set of dialogue contexts into a restricted number of dialogue context classes. The distance measure used in this clustering method is based on the difference of mutual information between classifications obtained in two consecutive algorithm steps. A classification represents the set of dialogue contexts classes.

Let $\mathbf{E}$, be the input alphabet, containing a sample of each event (trigram) existing in a given training database and $\mathbf{C_i}, \mathbf{C_{i+1}}$ the classifications obtained at the steps $\mathbf{i}$ and $\mathbf{i+1}$ respectively:

$$\mathbf{E} = \{e_1, .., e_n\}$$
$$\mathbf{C_i} = \{c_1, .., c_{p-1}, c_p, c_{p+1}, .., c_{q-1}, c_q, c_{q+1}, .., c_m\}$$
$$\mathbf{C_{i+1}} = \{c_1, .., c_{p-1}, c_{p+1}, .., c_{q-1}, c_{q+1}, .., c_m, c_r\}$$

At step $\mathbf{i+1}$ class c_r is obtained by merging the classes c_p and c_q, that means that the dialogue context class c_r will contain the sum of dialogue context previously contained in c_p and c_q.

Using the definition of the mutual information, we obtain for the difference of mutual information the following expression:

$$\mathbf{dI_i^{i+1}} = -\mathbf{H(E|C_{i+1})} + \mathbf{H(E|C_i)}$$

Using the conditional entropy $\mathbf{H(E|C_i)}$ [1] as follow:

$$\mathbf{H(E|C_i)} = \sum_{k=1}^{Card(E)} \sum_{j=1}^{Card(C_i)} \frac{n_j^k}{N} \cdot log_2 \frac{n_j^k}{N_j}$$

[1]where n_j^k is the number of occurrences of the event e_k in the class c_j, N_j is the whole number of events existing in class c_j and N is the global number of events (N is constant for a given database)

we obtain the following expression for the variation of mutual information:

$$\mathbf{dI_i^{i+1}} = \sum_{k=1}^{Card(E)} (\frac{n_r^k}{N} \cdot log_2 \frac{n_r^k}{N_r} - \frac{n_p^k}{N} \cdot log_2 \frac{n_p^k}{N_p} - \frac{n_q^k}{N} \cdot log_2 \frac{n_q^k}{N_p})$$

Our algorithm starts with single element clusters, i.e. a classification where each class contains only one dialogue context. The dialogue contexts that occur in the training database less then an established threshold (a number we approximate to 5) were eliminated from the initial classification. For these dialogue contexts we use during the recognition the context-independent LM, that is trained using the whole database.

Since we can demonstrate that:

$$\mathbf{dI_i^{i+1}} < 0$$

we obtain for the initial classification the maximal value for the mutual information (I_{max}). The goal of our algorithm is to merge as much classes as possible, with the minimal decrease of mutual information. We merge, thus, at each step the two classes that provide the minimal value for $\mathbf{abs(dI_i^{i+1})}$.

The merge process could continue till all classes are merged into a single class. For this single class classification, we obtain the minimal value for the mutual information (I_{min}). We must choose a classification that assures a good trade off between the value of the mutual information and the number of classes.

Many selection criteria have been included and tested in our algorithm:

a) Stop the algorithm when a given number of classes is reached.

b) Use of the perplexity value [2] as measure for the best classification.

c) Stop the algorithm when a given percentage (β) of the maximal variation of mutual information, ($\mathbf{dImax}$) (the difference between I_{max} and I_{min}) is reached. That means that the algorithm proceeds until the following condition is reached, for the step s:

$$\mathbf{abs(dI_0^s)} > \beta \cdot \mathbf{dImax}$$

By analizing the experimental results of different applications, we have observed that two phases can be distinguished in the behavior of the perplexity value at each algorithm step. In the first phase, the perplexity value can be considered constant, while in the second one a fast increase can be observed (see the representation of the perplexity values at the top of Figure 2).

We consider optimal the classifications near the end of the first phase, because they present a low value for the perplexity with a limited number of classes. Moreover, the optimal classifications correspond to a mutual information decrease of about 5% (a quite acceptable decrease). Therefore, we decided to use the third one as our default selection criterion, with β set to 5%.

Even if the second criterion guarantees the optimum of the perplexity value, the classification obtained with our default criterion is a very good approximation of the best one. Its main advantage is that it does not need any development database for computing the perplexity. In Figure 2 we present a plotting of the perplexity and

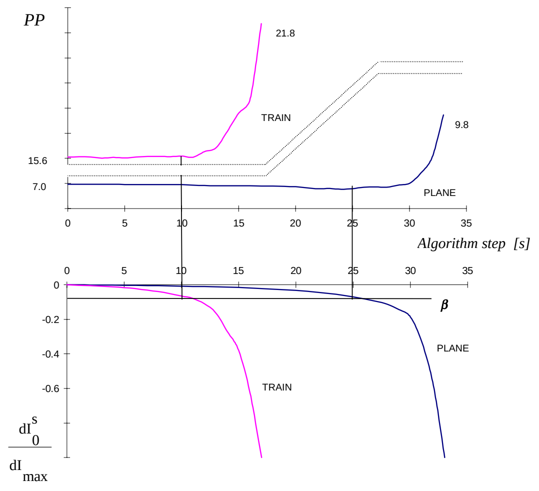

Figure 2: Perplexity and mutual information at each algorithm step.

mutual information values as a function of the current algorithm step, for two different applications. The first one is the Train Time Table Scheduling application (labeled by TRAIN), while the other one is the Flight Time Table Scheduling application (labeled by PLANE). For the Train Time Table Scheduling application, we used the database DIALOGOS. This database contains the transcription of more then 1,400 dialogues made by 500 naive users. 13,138 utterances are used for training and 1694 utterances are used for testing. For the Flight Time Table Scheduling application, we used a database that contains written sentences acquired during the validity test of the dialogue strategy. We divide this database into 6,239 training utterances and 3,000 testing utterances.

The classification algorithm for the first application starts with 17 classes only and stops after 10 merges, while the second begins with 33 classes and finds the optimum after 25 steps.

The presented algorithm allows the full automatic creation of the context-dependent LMs. Its computational complexity is $\mathbf{O(N \cdot n^2)}$, where N is the global number of events in the training database and n is the initial number of classes.

4. AUTOMATIC vs. MANUAL CLASSIFICATION

For the creation of the manual classification described in detail in [12] the following general heuristics, based on the structure of a dialogue context, were used:

- the dialogue context with different DAs are separated, while the composed dialogue act REQUEST-AND-CONFIRMATION is considered as a REQUEST DA, since the confirmation is implicit.

- the dialogue contexts with the same DA, are subsequently divided using the parameters of the DA. So,

[2] Also other measures could be used, like the word accuracy.

we kept together only the parameters that represent the same concept, like the confirmation of part_of_day, hour or time_interval in case of time expression.

Observing the results of the automatic classification, we see that the requests and the confirmations are always kept separated. Moreover, the REQUEST-AND-CONFIRMATION DAs were actually merged with the corresponding REQUEST ones.

Another common characteristic between manual and automatic classification is that, usually, the parameters representing the same concept are kept into the same class. For example, the confirmations for day_and_month, relative_day, week_day are clustered into the same class (the class for confirmations of the same concept: date). One exception is made by the confirmation of part_of_day: in case of the manual classification it is merged with the confirmation of hour, while the automatic classification put it with confirmations of the date concept.

After merging all the confirmations into one class, many of the request classes still remain separated. This classification produces a set of specific LMs, that maintains a quite reduced perplexity value, while continuing the algorithm until a single class is obtained, we notice a significant increment of perplexity. The above described behavior of the automatic clustering confirms another result already presented in [12]: the use of the context-dependent LMs improves mostly the recognition performances of the system with respect to the requests. Comparing the performances at the perplexity and word accuracy levels (Table 1) we see that the automatic classification does not give a significant improvement. This is not surprising because the manual and automatic classifications are actually very similar.

	PP	WA %
Manual classification	15.7	76.4%
Automatic classification	15.6	76.6%

Table 1: Perplexity and Word Accuracy Results

The main advantage of the automatic classification algorithm is that it makes possible the full automatic creation of context-dependent LMs, without any loss of performance. This aspect is very important for fast prototyping of new spoken dialogue systems.

5. CONCLUSIONS

The above described automatic classification algorithm has been developed for the clustering of dialogue contexts towards the creation of context-dependent dialogue LMs. The results of the automatic clustering are compared with a manual classification described in [12] and they are actually very similar. Therefore the automatic clustering algorithm can be used for the creation of a set of context-dependent LMs for spoken dialogue applications.

The automatic clustering algorithm has already been integrated into the development system for the Dialogos spoken dialogue system [1] and the use of context dependent LMs has been tested on two spoken dialogue applications: the Train Time-table Scheduling System and the Flight Time-table one.

6. REFERENCES

[1] Albesano D., P. Baggia, M. Danieli, R. Gemello, E. Gerbino, C. Rullent, "A Robust System for Human-Machine Dialogue in Telephony-Based Applications", *International Journal of Speech Technology*, Vol 2, No 2, 101-111, 1997.

[2] Andry F., "Static and Dynamic Predictions: A Method to Improve Speech Understanding in Cooperative Dialogues", *Proceedings of ICSLP*, Banff, Canada, 639-642, 1992.

[3] Baggia P., C. Rullent, "Partial Parsing as a Robust Parsing Strategy", *Proceedings of ICASSP*, Minneapolis, Minnesota, Vol 2, 123-126, 1993.

[4] Baggia P., M. Danieli, E. Gerbino, L. Moisa, C. Popovici, "Contextual Information and Specific Language Models for Spoken Language Understanding", *Proceedings of SPECOM*, 51-56, October 1997.

[5] Drenth E. W., B. Rueber, "Context-dependent probability adaptation in speech understanding", *Computer Speech and Language*, Vol 11, 225-252, 1997.

[6] Eckert W., F. Gallawitz, H. Niemann, "Combining Stochastic and Linguistic Language Models for Recognition of Spontaneous Speech", *Proceedings of ICASSP*, Atlanta, Vol 1, 423-427, 1996.

[7] Gemello R., D. Albesano, F. Mana, "Continuos Speech Recognition with Neural Networks and Stationary-Transitional Acoustic Units", *Proceedings of the IEEE International Conference on Neural Networks*, Houston, USA, Vol 4, 2107-2111, 1997.

[8] Godfrey J., E. Holliman, J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development", *Proceedings of ICASSP*, 517-520, 1992.

[9] Jurafsky D., R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolke, P. Taylor, C. Van Ess-Dykema, "Switchboard discourse language modeling project report", *Technical report*, center for Speech and Language Processing, Johns Hopkins University, Baltimore.

[10] Niedermaier G., "Linguistic modeling in the context of oral dialogue", *Proceedings of ICSLP*, Banff, Canada, 635-638, 1992.

[11] Placeway P., R. Schwarz, P. Fung, L. Nguyen, "The Estimation of Powerful Language Models from Small and Large Corpora", *Proceedings of ICASSP*, Minneapolis, Minnesota, Vol 1, 73-76, 1993.

[12] Popovici C., P. Baggia, "Specialized Language Models Using Dialogue Predictions", *Proceedings of ICASSP*, Muenchen, Germany, Vol 2, 815-818, Apr 1997.