Gefördert vom

bmb+f

Bundesministerium für
Bildung, Wissenschaft,
Forschung und Technologie

Verbmobil

Verbundvorhaben

Predicting Dialogue Acts for a Speech-To-Speech Translation System

Norbert Reithinger, Ralf Engel,
Michael Kipp, Martin Klesen

DFKI GmbH

Vm

Report 151
August 1996

August 1996

Norbert Reithinger, Ralf Engel, Michael Kipp, Martin Klesen

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
Stuhlsatzenhausweg 3
66123 Saarbr"ucken

Tel.: (0681) 302 - 5346
Fax: (0681) 302 - 5341
e-mail: `Reithinger@dfki.uni-sb.de`

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01IV101K/1 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

Abstract

We present[1] the application of statistical language modeling methods for the prediction of the next dialogue act. This prediction is used by different modules of the speech-to-speech translation system VERBMOBIL. The statistical approach uses deleted interpolation of n-gram frequencies as basis and determines the interpolation weights by a modified version of the standard optimization algorithm. Additionally, we present and evaluate different approaches to improve the prediction process, e.g. including knowledge from a dialogue grammar. Evaluation shows that including the speaker information and mirroring the data delivers the best results.

1 INTRODUCTION

VERBMOBIL is a system for the translation of spontaneous speech in face-to-face situations, mainly from German to English [6] (c.f. `http://www.dfki.uni-sb.de /verbmobil`). The system consists of more than 20 modules for speech recognition, linguistic analysis, context processing, generation, and speech synthesis.

The dialogue module stores data about the dialogue context and provides this information to the other modules in the system. Dialogue processing is based on so called dialogue acts. For each utterance in the system, a dialogue act is computed, either using linguistic or statistic methods. We use 42 acts that describe both the intention and partly the propositional content of an utterance. They are organized in a hierarchy with additional 18 acts describing primarily intentions at a domain independent level, like `suggest`, `init`, and `accept`. The results presented in section 3 are computed using these 18 dialogue acts.

Within the dialogue module, we use both statistical and knowledge based methods to represent and process the dialogue context [1, 4]. The main components are the statistical dialogue act prediction which is described in this article, the plan recognizer, and the dialogue memory.

The empirical basis of our work is the VERBMOBIL corpus which consists of over 1000 spoken scheduling dialogues that have been recorded and transliterated. Over 300 of them were manually tagged with dialogue acts. This data is used as training and test material for main parts of the dialogue component

The prediction of dialogue acts is used by various system modules. For example, semantic evaluation uses them to focus the algorithm for the determination of the next utterance's dialogue act. Another module that uses this information heavily is a robust information extraction module.

In the remainder of this paper we first present the basic prediction algorithms, together with some modifications. We then evaluate the different methods and show which one delivers the most reliable results.

2 PREDICTION ALGORITHMS

2.1 Statistical Background

The task to be solved consists of predicting the next dialogue act in an ongoing conversation. Since this problem is almost identical to the task of predicting the

[1] This paper was accepted for ICSLP-96

next word in a sentence considering all previously uttered words, we can apply well-known language model techniques from the field of speech recognition [2]. Instead of processing the words of a text or a dialogue, the dialogue acts describing the content are the basic processing units.

When processing a dialogue previously uttered dialogue acts which are available as *history* can be used. The most probable following dialogue act d_j is the one satisfying

$$P(d_j|\, d_1, \ldots, d_{j-2}, d_{j-1}) = \max_d \; P(d|\, d_1, \ldots, d_{j-2}, d_{j-1}),$$

if $d_1, \ldots, d_{j-1}$ represent the history of formerly uttered dialogue acts. Since it is not possible to determine the probabilities of arbitrarily long sequences of dialogue acts, we have to approximate it. From language modeling we can apply the *deleted interpolation* method using n-grams [2]. An n-gram is a sequence of n subsequentially uttered dialogue acts $(d_i, \ldots, d_{i+n-1})$ and serves as a shortened history, including also the dialogue act d_i whose probability is to be calculated. The n-gram probability $P(d_j|\, d_{j-n+1}, \ldots, d_{j-1})$ approximates the required probability $P(d_j|\, d_1, \ldots, d_{j-1})$. Since even short histories very often are not in a training set, the probability is interpolated by combining histories of different lengths n. Each probability is multiplied by a fixed *weight* q_i.

The n-gram probabilities P are approximated by the n-gram *relative frequency* f_n, which are simply the number of occurrences of the respective n-gram, say $(d_1, \ldots, d_n)$, in the training corpus, divided by the number of occurrences of the $(n-1)$-gram $(d_1, \ldots, d_{n-1})$. Inserting frequencies f_i in the above formula we gain

$$P(d_j|\, d_{j-N+1}, \ldots, d_{j-1}) = \sum_{n=1}^{N} q_n\, f_n(d_j|\, d_{j-n+1}, \ldots, d_{j-1})$$

as the formula used to approximate $P(d_j|\, d_1, \ldots, d_{j-1})$, thus being the basis of all further computation.

Starting from a corpus from which we can get the frequencies, the problem of how to determine the weights q_i must be solved. First, we present a method that computes those frequencies before the dialogue is processed. We then show a technique which works dynamically and adapts the weights during dialogue analysis. Finally, we introduce some other ideas of how to possibly enhance performance.

2.2 Determining the Model Weights

2.2.1 The Markov Chain Method

The non-negative weights in the n-gram probability estimation satisfy $\sum_{k=1}^{N} q_i = 1$. They should also fulfill the *maximum-likelihood criterion*, i.e. they are adjusted to maximize the probability $P(S)$ of the observed data.

The general idea is to model the dialogue act generating process with a *hidden Markov chain* in which the weights q_i appear as transition probabilities between some of the states [2] . Then a simplified version of the well known *forward backward algorithm* [5] can be used to carry out the desired optimization.

We divide the annotated dialogues in two disjoint sets of training and test data. The frequency tables are built with the training dialogues, while the optimization is carried out using the test dialogues. Besides that we generalize the idea of dialogue act sequences, allowing $S = (d_1, d_2, \ldots, d_n)$ to denote multiple dialogues, i.e. sequences of dialogue acts separated by a special end marker. Stepping through S means in this case stepping through each of the single dialogues in turn. Now if S is such a generalized sequence of dialogue acts and $\mathcal{L} = \{v_1, v_2, \ldots, v_r, \epsilon\}$ is the set of all *different* dialogue acts occurring in S, augmented by the empty act ϵ, then we construct a hidden Markov model as proposed in [2]

For each history $\mathcal{S}_i = (d_{i-N+1}, \ldots, d_{i-1})$ of length $N-1$ that occurs in S, there are $N+1$ different states, namely $Z_0[\mathcal{S}_i], \ldots, Z_N[\mathcal{S}_i]$. The *null transitions* $Z_0[\mathcal{S}_i] \longrightarrow Z_k[\mathcal{S}_i]$—i.e. transitions after which the invisible word ϵ is generated—take place with probability q_k and the transitions from $Z_k[\mathcal{S}_i] = Z_k[d_{i-N+1}, \ldots, d_{i-1}]$ into the r successive states $Z_0[d_{i-N+2}, \ldots, d_{i-1}, v_l]$ in which the dialogue act v_l is produced are chosen according to the k-gram probabilities $f_k(v_l \mid d_{i-k+1}, \ldots, d_{i-1})$ provided by our statistics. Since it is a hidden model it is not possible to observe which of the k transitions was taken at any time. On the other hand for any position in S where the sequence $\mathcal{S}_i$ appears followed by some dialogue act v_l, we can compute the *probability* $p_k(v_l)$ that the transition $Z_0[\mathcal{S}_i] \longrightarrow Z_k[\mathcal{S}_i]$ took place just before v_l was generated. In fact,

$$p_k(v_l) = \frac{q_k f_k(v_l \mid d_{i-k+1}, \ldots, d_{i-1})}{P(v_l \mid d_{i-N+1}, \ldots, d_{i-1})} \quad \text{and} \quad \sum_{k=1}^{N} p_k(v_l) = 1 \quad .$$

As a consequence, the probabilities $p_k(v_l)$ can be seen as the fraction of times the respective transition is taken. They are then used to compute estimates for the actual counts.

2.2.2 Smoothing the Weights

When implementing the algorithm described above, we observed that the new, re-estimated weights $\hat{q}_k$ were almost always totally different from the previous ones, sometimes leading to a lower probability $P(S)$ and what was even worse, there was no convergence towards some "final" weights. An analysis of what could have caused this problem revealed that more than eighty percent of all sequences $\mathcal{S}_i$ occurred only once or twice in S leading to a bad estimation of the $\hat{q}_k$ which is based on computing relative frequencies.

To stabilize the algorithm we use *global* counts $\hat{C}(Z_0), \hat{C}(Z_0 \rightarrow Z_1), \ldots, \hat{C}(Z_0 \longrightarrow Z_N)$ to re-estimate the weights instead of the local weights which are based on a specific history $\mathcal{S}_i$. First we construct a list $\mathcal{H}$ of all *different* histories of length $N-1$ occurring in S and sort it by the number of their occurrence. Then we step through $\mathcal{H}$, increasing the global counts by the respective local ones and re-estimate the weights before choosing the next history.

To compare the performances of the original algorithm and our modified version, we let them both run on the same training/test corpus, using up to 4-grams in the interpolation. Figure 1 illustrates the development of the weights during the re-estimation process. Initially, the q_k, $k = 0 \ldots 3$, along the y-axis have the same value. The x-axis shows the different histories processed by the algorithm. As can be seen in the left diagram the q_n oscillate without converging to usable values. The right diagram shows the development of the interpolation values when using the smoothing technique.

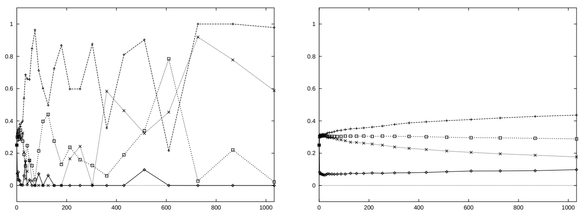

Figure 1: Weights re-estimated with the original version (left) and the smoothed version (right)

2.2.3 Dynamic Adaptation

If, in the process of prediction, the incoming series of dialogue acts is very different in structure from the test corpus used for the Markov chain algorithm, unsatisfactory results are to be expected. Of course, after one such case happening the Markov algorithm could be applied again but this does not guarantee avoidance of similar cases in the future.

Therefore a dynamic adaptation approach is desirable where the weights can be adapted depending on the performance of the single k-grams in the course of processing the current dialogue act series. Again, a technique used in language modeling proves useful here (see e.g. [3]).

The adaptation is conducted by changing the weights each time a dialogue act has been processed. We call this an iteration step for a weight q_k. The formula for one iteration step from q_k to $\tilde{q}_k$ (originally taken from [3]) is

$$\tilde{q}_k = q_k \sum_{m=n-L}^{n-1} \frac{1}{L} \left(\frac{f_k(s_m \mid s_{m-k+1}, \ldots, s_{m-1})}{P(s_m \mid s_{m-N+1}, \ldots, s_{m-1})} \right)$$

The calculation takes a history of L dialogue acts into account that were produced just before the prediction of the next dialogue act has to be made. It compares the performance of the different relative frequencies, rewarding good performance by increasing the respective weight and penalizing bad results accordingly by decreasing the respective weight.

Dynamic adaptation makes sense only if the different n-grams actually have a different quality in terms of prediction accuracy for different parts of a dialogue (e.g. the bigram might be best for beginning and end of a dialogue and the trigram best for the middle part). Experimental results show that this is indeed the case.

2.3 Including a Dialogue Grammar

To enhance the prediction performance we also examined whether declarative knowledge sources, like a dialogue grammar, can be included in the prediction process . The dialogue component comprises a dialogue grammar which describes at what "stage" the dialogue currently is, like e.g. starting phase, end, proposal or reaction. It is encoded as an automaton with six states, where dialogue acts are at the edges between the states.

An obvious idea to exploit such a knowledge source is to train the grammar. i.e. attribute probabilities to the states and transitions, and use this knowledge for prediction. When evaluating this method the performance was always 5 to 15 percent worse than the purely statistical approach. Therefore, we dropped it entirely [4].

We investigated two different ways to include knowledge from this grammar into the interpolation formula. The first one is the extension of the interpolation formula with an additional weight q_a for the automaton:

$$\begin{aligned} P(d_i \mid d_{i-N+1}, \ldots, d_{i-1}) &= \sum_{k=1}^{N} q_k \, f_k(d_i \mid d_{i-k+1}, \ldots, d_{i-1}) \\ &\quad + q_a \, f_a(d_i \mid cs) \end{aligned}$$

where $f_a(d_i \mid cs)$ is the probability of d_i under the condition that the automaton is in state cs.

The second way to integrate the automaton is to replace older dialogue acts in the history by the corresponding automaton states. N-grams with order $n > 3$ usually do not contribute significantly in the interpolation, since many dialogue act sequences occur in the test set but not in the training set. Using automaton states instead of dialogue acts reduces this effect, since there are less automaton states than dialogue acts and therefore a clustering of dialogue acts is achieved.

The interpolation formula for e.g. $N = 3$ then looks like

$$\begin{aligned} P(d_i \mid d_{i-2}, d_{i-1}) &= q_1 \, f_1(s_i) + q_2 \, f_2(d_i \mid d_{i-1}) \\ &\quad + q_3 \, f_3(d_i \mid \mathbf{a_{i-2}}, d_{i-1}) \end{aligned}$$

where a_{i-2} is the automaton state after processing d_{i-2}.

2.4 Exploiting the Scenario

In our scenario of face-to-face dialogues it is known which of the two speakers made a contribution. Therefore, we can augment the dialogue act with a tag for the speaker.

This information can also be exploited for our prediction task. If e.g. speaker A poses a question and the second utterance is brought forth again by A, one could expect this utterance to be an explanation, correction or an additional question. If, however, the second utterance is produced by speaker B, it is most probably a reply. This demonstrates the potential value of the directional information for the prediction. In our scenario with two speakers this means to duplicate the number of dialogue acts, for example to replace `reject` with `reject-ab` or `reject-ba`, depending on the direction.

Having integrated a mechanism for taking into account speaker information, we realized that we could duplicate the number of training dialogues by "mirroring" them. That is, for each dialogue we created a counterpart by exchanging the speaker information.

3 EVALUATION RESULTS

The algorithms described above have been implemented as part of a flexible workbench. Using the annotated corpus, we experimented with the various approaches in order to get the best prediction results.

A first observation was that prediction hit rates, i.e. correct predictions of the following dialogue act, vary only in a limited bandwith regardless which dialogues are used for training and test. When using the 18 intentional acts hit rates are are about 40% when predicting only the best dialogue act, about 65%, when predicting two, and about 75% when predicting three dialogue acts. We speak of a a correct prediction when the actual act was one of the two or three acts predicted.

The hit rate drops by about 10% when using all 42 dialogue acts. Dialogue act perplexity using all acts is about 11. In all tests only for n-grams $n < 5$ the q_n determined by the Markov Chain Method have a significant value. Using the 42 acts, also 4-grams do not contribute significantly to the interpolation.

method	*Markov*	*dyn. adapt.*
plain	72.24%	71.37%
speaker information	75.49%	74.92%
speaker info.+mirror	76.05%	75.83%
automaton (v1)	75.32%	73.63%
automaton (v2)	75.82%	n/a

Figure 2: Prediction results for three predictions

To demonstrate the influence of the approaches presented, we selected 150 annotated dialogues and divided them into training and test data. We used about 70% of the dialogues as training and the rest as test data. For the experiments the 18 intentional acts were used. The interpolation was done using up to 4-grams, and hit rates for three predictions were tested.

Fig. 2 shows the hit rates for the different algorithms. We get the best results when we include speaker information in the algorithm and mirror the dialogues. The inclusion of the automaton either by adding an additional factor in the formula (v1) or by replacing elements of the dialogue act history (v2) also does not improve the results.

The addition of the dynamic adaptation of the weights yields worse prediction results for all methods tested. This observation could be made in other experiments as well. The adaptation obviously reintroduces the effects of the unsmoothed Markov algorithm. Since many, even relatively short sequences of dialogue acts occur only rarely in the dialogues, an overadaptation of the interpolation weights to the most recent input takes place.

4 CONCLUSION

For the prediction of dialogue acts in VERBMOBIL we adopted a statistical approach from language modeling, namely using deleted interpolation to compute the probability of a sequence of dialogue acts. The original approach to compute the interpolation weights has been modified to get converging values. In addition to the standard method, we looked for additional options in order to improve the prediction accuracy. For example, we adapted the weights during the runtime of the system and we included other knowledge sources in our statistical model. However, from all the ideas presented, only the inclusion of the speaker information combined with mirroring the data resulted in a significantly better performance in the current application.

The implemented system has been in use for more than a year now. It can be – and has been – easily adapted to different sets of dialogue acts from different scenarios. The various methods implemented are continuously being evaluated to select those which deliver the best overall performance.

References

[1] Jan Alexandersson, Elisabeth Maier, and Norbert Reithinger. A Robust and Efficient Three-Layered Dialogue Component for a Speech-to-Speech Translation System. In *Proceedings of the 7th Conference of the European Chapter of the ACL (EACL-95)*, pages 188–193, Dublin, Ireland, 1995.

[2] Fred Jelinek. Self-Organized Language Modeling for Speech Recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, 1990.

[3] Reinhard Kneser and Volker Steinbiss. On the dynamic adaptation of stochastic language models. In *Proceedings ICASSP-93*, pages 585–589, 1993.

[4] Norbert Reithinger and Elisabeth Maier. Utilizing Statistical Speech Act Processing in VERBMOBIL. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Cambridge, MA, 1995.

[5] Ernst G. Schukat-Talamazzini. *Automatische Spracherkennung. Grundlagen, statistische Modelle und effiziente Algorithmen.* Artificial Intelligence - Künstliche Intelligenz. Vieweg Verlag, Braunschweig - Wiesbaden, 1994.

[6] Wolfgang Wahlster. First Results of Verbmobil : Translation Assistance for Spontaneous Dialogs. In *Proceedings of ATR International Workshop on Speech Translation (IWST'93). Kyoto, Japan, November 8–9*, 1993.