In Proc. EUROSPEECH '97, Rhodes, Vol. 1, pages 207–210

INTEGRATED DIALOG ACT SEGMENTATION AND CLASSIFICATION USING PROSODIC FEATURES AND LANGUAGE MODELS

V. Warnke[1], R. Kompe[2], H. Niemann[1], E. Nöth[1]

[1]Universität Erlangen–Nürnberg, Lehrstuhl für Mustererkennung, 91058 Erlangen, Germany
http://www5.informatik.uni-erlangen.de/
[2]Sony International (Europe) GmbH, 70736 Fellbach, Germany

Abstract

This paper presents an integrated approach for the segmentation and classification of dialog acts (DA) in the VERBMOBIL project. In VERBMOBIL it is often sufficient to recognize the sequence of DAs occurring during a dialog between the two partners. In our previous work [5] we segmented and classified a dialog in two steps: first we calculated hypotheses for the segment boundaries and decided for a boundary if the probabilities exceeded a predefined threshold level. Second we classified the segments into DAs using semantic classification trees or stochastic language models. In our new approach we integrate the segmentation and classification in the A^*–algorithm to search for the optimal segmentation and classification of DAs on the basis of word hypotheses graphs (WHGs). The hypotheses for the segment boundaries are calculated with the help of a stochastic language model operating on the word chain and a multi-layer perceptron (MLP) classifying prosodic features. The DA classification is done using a category based language model for each DA. For our experiments we used data from the VERBMOBIL-corpus.

1. INTRODUCTION

VERBMOBIL is a speech-to-speech translation project [1] in the domain of appointment scheduling, i.e. two persons try to fix a meeting date, time, and place. Usually both dialog partners will speak English. If they do not know how to express themselves they can switch to their mother tongue and the VERBMOBIL-system starts translating, after a command is given to the system. To keep track of the dialog it is necessary for the system to know the state of the dialog at any time. This is done in terms of dialog acts (DAs) as one of the tasks of the dialog module within VERBMOBIL. DAs are, e.g., “greeting”, “confirmation of a date”, “suggestion of a place”. In VERBMOBIL one turn of a dialog often consists of more than one DA. DAs are detected on the basis of WHGs using statistical classifiers. Previously, the processing was done suboptimally within two steps: first, the utterance is segmented into DA units (DAUs). Second, these units are classified into DA categories (DACs). In this approach we integrated the segmentation and classification task in an A^*-search. Thus we have to define a cost-function, which uses estimations of probabilities for segment boundaries, DAs and the dialog model, i.e., the sequence of dialog acts. The probabilities for segment boundaries are estimated with a MLP on the basis of prosodic feature vectors computed for each of the word hypotheses in the WHG [4]. All the other estimations for the cost-functions of the A^*–algorithm are calculated during the search based on n–gram language models using the word chain from the WHG. We use one language model for each DA, one to model the sequences of the DAs and one to decide for segment boundaries on the word chain.

2. DIALOG ACTS IN VERBMOBIL

In VERBMOBIL the whole dialog of two persons is seen as a sequence of DAs, which means that DAs are the basic units on the dialog level. The DACs are defined according to their illocutionary force, e.g., ACCEPT, SUGGEST, REQUEST, and can be sub categorized as for their functional role or their propositional content, e.g., DATE or LOCATION depending on the application. In the VERBMOBIL domain 18 DACs are defined on the illocutionary level with 42 sub categories [3]. In Figure 1 each example shows one turn hand–segmented into DAUs and hand–labeled with the appropriate DAC. Each DAU corresponds to one (cf. example 2) or more (cf. example 1) DA(s). Since in spontaneous speech many incomplete and incorrect syntactic structures occur, e.g., a lot of elliptic sentences or restarts, it is not easy to give a quantitative and qualitative definition of the term DA. In VERBMOBIL, criteria were defined for the manual segmentation of turns based on their textual representation and for the manual labeling of these segments with DACs [6]. No prosodic information is used for labeling, in order to be able to label the dialogs without having to listen to them. Thus it was possible to reduce the labeling effort. Nevertheless, we will prove in Sections 4.3. and 4.4. that for the automatic detection of DAUs prosodic markers are very important cues, cf. also [2]. These manually created labels are used as reference for the training and evaluation of our stochastic models, as described in the following section.

3. METHODS USED

3.1. Multi–layer Perceptrons

Multi–layer perceptrons were trained to recognize the DA–boundaries in a similar way as the prosodic phrase boundaries described in [4]. For each word–final syllable a vector of prosodic features is computed

*This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01 IV 102 H/0. The responsibility for the contents lies with the authors.

Ex. 1	uh Matt this is Brian here again	INTRODUCE_NAME
	I have to meet you sometime uh uhm this month to uh discuss the documentation for the code you have written	SUGGEST_SUPPORT_DATE, MOTIVATE_APPOINTMENT
Ex. 2	well I have a meeting all day on the thirteenth	SUGGEST_EXCLUDE_DATE
	and on the fourteenth I am leaving for my bob sledding vacation until the nineteenth	SUGGEST_EXCLUDE_DATE
	uh how 'bout the morning of the twenty second or the twenty third	SUGGEST _SUPPORT_DATE

Figure 1: Two turns segmented into DAUs and labeled with the respective DACs.

automatically from the speech signal. This vector models prosodic properties over a context of six syllables taking into account duration, pause, F0–contour and energy. This is based on a time alignment of the phoneme sequence corresponding to the spoken words. The MLP has one output node for the DA boundaries (D) and one for the other word boundaries (¬D).

We assume that the MLP estimates posterior probabilities. However, in order to balance for the a priori probabilities of the different classes, during training the MLP was presented with an equal number of feature vectors from each class. The best classification result so far (cf. below) was obtained with 117 prosodic features for each word–final syllable and an MLP with 60/30 nodes in the first/second hidden layer.

3.2. Polygram Language Models (LM)

A certain kind of n–gram language models – so called polygrams [8] – are used for the segmentation and classification of DAs. Polygrams are a set of n–grams with varying size of n. They are superior to standard n–gram models because n can be chosen arbitrarily large and the probabilities of higher order n–grams are interpolated by lower order ones. The interpolation weights are optimized using the EM algorithm. There are several interpolation methods possible for the polygrams, which are described in detail in [8, 9]. In this paper we used the polygrams to model three different stochastic processes; we set $n = 5$ for DA classification, $n = 3$ for DAU segmentation, and $n = 2$ to model the DAC sequences.

Segmentation into DAUs

For the *segmentation* of turns into DAUs we trained LMs, which model the probability for the occurrence of a boundary after the current word given the neighboring words, cf. [4]. For each word boundary, symbol sequences $\dots w_{i-2}w_{i-1}w_iv_iw_{i+1}w_{i+2}\dots$ are considered, where w_i denotes the i-th word in the spoken word chain and v_i is either D or ¬D. Note that theoretically, we should model sequences $\dots w_{i-1}v_{i-1}w_iv_iw_{i+1}v_{i+1}\dots$; experiments showed, however, that this yields worse results. In this case the polygram obviously is not able to cover a sufficiently large word context.

Classification of DACs

We used polygram language models for the *classification* of different DACs. For each of the 18 illocutionary DACs a separate *category* based LM is trained on the corresponding word sequences obtained from the hand–segmented and hand–labeled turns. For the interpolation of the higher order n–grams, we use a new rational interpolation scheme as presented in [9]. During classification integrated in the A^*–search we estimate the probabilities $P(w_m \mid w_{m-n}\dots w_{m-1}) \approx C(w_m) \cdot Q(w_{m-n}\dots w_m)$ for the current DAC using the interpolation scheme

$$Q(w_{m-n}\dots w_m) =$$

$$\frac{\sum_{i=1}^{n} p_i \cdot (1/L)^{n-i} \cdot \#_i(\mathcal{C}(w_{m-n})\dots\mathcal{C}(w_{m-1})\mathcal{C}(w_m))}{\sum_{i=1}^{n} p_i \cdot (1/L)^{n-i} \cdot \#_i(\mathcal{C}(w_{m_n})\dots\mathcal{C}(w_{m-1}))},$$

where $\#_i$ counts the i predecessors of the category sequence $\mathcal{C}(w_{m-n})\dots\mathcal{C}(w_{m-1})$, $\mathcal{C}(w_i)$ returns the category for the given word w_i, L is the lexicon size and p_i are the interpolation coefficients. The probability for w_m belonging to the category $\mathcal{C}(w_m)$ is computed using the *word emission probability*

$$C(w_m) = \frac{\#(w_m) + 1}{\#(\mathcal{C}(w_m)) + C_S * 1},$$

where # represents the frequencies of occurrence of w_m and of $\mathcal{C}(w_m)$ in the training corpus. C_S is the number of categories used in the LM. A detailed description of the interpolation methods is given in [9].

Modeling DAC Sequences

In the VERBMOBIL-project a *dialog model* was defined by a finite state automaton [3]. Here, we use a polygram language model to compute the probability for the DAC sequences. For the training task we used the hand–labeled DACs of each turn in the training corpus in order to build DAC sequences $D_1D_2\dots D_m$, where D_i is one of the 18 DACs (e.g., "GREET INTRODUCE INIT SUGGEST "). Using these sequences we trained and validated the LM. For the classification within the A^*-search, we compute the n predecessor DACs and calculate the probability of the actual DAC. We also used the above rational interpolation scheme to calculate $P(D_m \mid D_{m-n}\dots D_{m-1})$, but we did not use categories. Thus, $\mathcal{C}(\cdot)$ is the identity function and the function $C(\cdot)$ returns 1.

3.3. The A^*–Algorithm

In the following we will introduce informally the search procedure. The search proceeds left-to-right through a word graph in the general case; note, however, that in the following a word chain is considered as a linear word graph. A node of the search tree is defined by

- a path in the word graph starting at the first node in the word graph,
- a unique segmentation of the corresponding word chain into DAUs, and

```
Take the best scored search tree node from the agenda. Let $D'$, $W'$, and $L$ be the right-most DAC, word hypothesis, and the right-most word graph node.
FOR each word hypothesis $W$ which begins at $L$
    IF "$\neg$D" after word $W'$
    THEN Build a new search tree node, where "$W$ $\neg$D" is appended to the sequence of words and boundary symbols.
         Build a new search tree node, where "$W$ D" is appended to the sequence of words and boundary symbols and $D'$ is appended to the DAC sequence.
    ELSE FOR each DAC $D_i$
             Build a new search tree node, where "$W$ $\neg$D" is appended to the sequence of words and boundary symbols.
             Build a new search tree node, where "$W$ D" is appended to the sequence of words and boundary symbols and $D_i$ is appended to the DAC sequence.
```

Figure 2: Procedure for the expansion of a search tree node.

- a unique classification of the DAUs into DACs.

For the word chain of Ex. 1 (above), a possible node in a search tree could contain the following information:

uh ¬D Matt ¬D this ¬D is ¬D Brian ¬D here ¬D again D I ¬D have ¬D to ¬D meet ¬D
INTRODUCE_NAME, SUGGEST_SUPPORT_DATE

This means that the word chain is segmented into two DAUs with the boundary after again; the second DAU is not yet complete, because it is not bound by a D symbol to the right. The two DAUs are classified as indicated at the bottom of the table.

The search is based on the A^*–algorithm. At each step of the search, the optimal node of the search tree is taken from the agenda and it is expanded according to the algorithm presented in Figure 2. The successor nodes are built according to the possible successor words in the word graph and by additionally considering that the current DAU may continue, or that a new DAU may start. In the latter case, for each of the 18 DACs two different successors in the search space are created. Thus, for each successor word in the word graph 36 successor nodes in the search space are generated. These successors are then scored and inserted into the agenda.

The score integrates

- the scores of the different language models described above,
- the MLP score, as well as
- appropriate remaining costs.

During the search the scores can be efficiently computed in an incremental way. Note that the remaining costs have to be approximated using a fast Viterbi forward-backward search prior to the A^*–search. Since the different language models are suboptimal we found it appropriate to weight the individual scores before their combination. In this paper we apply the algorithm only to the spoken word chains. In this case, the search yields the optimal combined segmentation and classification of the word chain into DACs.

4. RESULTS

4.1. Data

All classification experiments were based on the same subsets of the German VERBMOBIL spontaneous speech corpus. For training, 96 dialogs (2459 turns of 57 different female and 58 male speakers, approx. 5.5 hours of speech) were considered; the test set comprises 31 dialogs (391 turns) of 20 different speakers (3 female, 17 male; approx. 1 hour of speech). The training set consists of 6496 and the test set of 992 DAs. For this paper we had to exclude 17 turns form the test set which contain DACs we are not able to model, because there were not enough representations in the training corpus. Thus, the results we achieved could not be compared with those presented in [5]. Therefore, we repeated the experiment of [5] on the new test corpus using the new language models; the results are presented in Table 1. So far, we evaluated our algorithm only on the spoken word chains.

4.2. Evaluation Procedure

With respect to the integration in the VERBMOBIL system, the DA classification has to deal with automatically segmented word sequences. For the evaluation, it has to be taken into account that DAUs may be deleted or inserted. Therefore, we align the recognized sequence of DAC class symbols with the reference for each turn. The alignment tries to minimize the Levenshtein distance. The percentage of correct (*corr*) classified DACs is given together with the percentage of deleted (*del*) and inserted (*ins*) segments in Table 1. Furthermore, the recognition accuracy (*acc*) measures the combined classification and segmentation performance; it is defined as $100 - subs - del - ins$, where *subs* denotes the percentage of misclassified DACs. Note that in this evaluation, a DA is considered as classified correctly if it is mapped onto the same DA category in the references; it does not matter if the segment boundaries agree with the hand–segmented boundaries. In this context the most important numbers are the correctly classified DAs versus the insertions. In the table results for different thresholds θ are given.

4.3. Subsequent Segmentation and Classification of Dialog Acts

In our previous approach we used in the first task the combination of MLP and LM for the segmentation. The DACs are classified in the second task using the LMs, as described in [5]. To compare our new results with those from our previous study, we repeated experiments with our new test set and better DAC-LMs as follows: First, we computed for each word boundary the probabilities $P(D)$ and $P(\neg D)$. Second we classified each boundary as D if $P(D) > \theta$ and as ¬D else. Third the word chains between each subsequent pair of D was extracted and classified with the LM into one out of the 18 DACs. In Table 1 results for different thresholds θ are given. The smaller θ the smaller the number of deleted segments and the larger the number of inserted segments. Note, that

θ	*acc* in [5]	*acc*	*corr*	*del*	*ins*
0.95	45.2	52.8	55.5	15.9	2.8
0.93	45.8	53.0	57.1	13.4	4.2
0.86	44.4	52.8	60.3	8.9	7.5
0.79	43.2	50.4	61.9	5.6	11.5

Table 1: Classification results for DACs using the two step approach

we improved the DA accuracy up to 8% using the new DAC–LMs (see Table 1).

4.4. Integrated Segmentation and Classification of DAs

In our new approach, we integrated segmentation and classification in the A^*–search as described in Section 3.3. Therefore it is not longer necessary to use a predefined threshold θ to segment the turns into DAUs. We calculate within the prosodic module hypotheses for the DAU boundaries using the MLP and attach the probabilities to each of the word hypotheses in the WHG. During the A^*–search the *DAU boundary probabilities* from the prosodic module are read from the WHG, the probabilities for the *18 DAC–LMs*, the *boundary–LM* and *DAC sequence–LM* are computed using the word chain from the WHG respectively the actual DAC sequence. For each expanded node in the WHG, the *costs* are computed by the *weighted sum* of the log. probabilities estimated by the MLP and the LMs, using the weights p_i, $i \in 1, 2, 3, 4$, as informally shown in

$$costs = p_1 \cdot \log P_{MLP} + p_2 \cdot \log P_{DACLM}$$

$$+ p_3 \cdot \log P_{sequenceLM} + p_4 \cdot \log P_{boundaryLM}.$$

Since segmentation and classification are integrated in the A^*–search, it is possible to overcome wrong boundary hypotheses from the prosodic module, using the estimations of the DAC and the segmentation language models, because the costs at the actual node become higher for the wrong boundary hypotheses and the path is not stored back at the top of the agenda. Thus we were able to reduce the insertion or deletion rate. Furthermore we achieved better correct and accuracy rates for all weight configurations using the integrated approach. We tested our new segmentation and classification system with varying weight configurations p_i (see above equation), but set the weight $p_4 = 2.0$ for all experiments. The results we achieved are given in Table 2. One can see, that the accuracy and correct rate improved, when we used a weight of $p_3 = 0.1$ for the DAC sequence model. At this time of our research we set the *remaining costs* to zero and had a full search, but the *real-time factor* of the system is less 1.8 for all weight configurations. In our future work we will examine an optimization algorithm for the weights p_i, using a validation set different from the test set.

5. CONCLUSION

The segmentation and classification of DAs is an important upcoming issue, because in real dialogs a turn can consist of more than one DA. Especially in the context of VERBMOBIL the segmentation and classification of DAs is necessary for keeping track of the dialog history. Previously we showed that DAs can be reliably classified based on automatically detected segments. In this paper we presented an algorithm for the integrated segmentation and classification of DAs. With this we were able to improve our DA recognition accuracy considerably. Note that our results cannot directly be compared to the DA recognition rates presented in [7], because in that paper the possible DAs for a DAU are restricted using information obtained over a sequence of turns, whereas we so far only work on a single turn. However, our algorithm can easily be extended for the application to sequences of turns.

p_1	p_2	p_3	*acc*	*corr*	*del*	*ins*
2.3	1.0	0.1	53.4	58.6	11.9	5.2
2.5	1.0	0.1	53.2	59.4	11.7	6.2
2.1	1.0	0.0	53.0	59.7	10.8	6.7
2.7	1.0	0.0	52.7	61.5	7.2	8.9

Table 2: Classification results for DACs using the integrated approach

In the future we will show that the algorithm is very well suited for the recognition of DAs on automatically recognized word hypotheses graphs. In fact the integrated segmentation and classification is the only useful approach for determining a DA sequence on the basis of a word hypotheses graph. Preliminary experiments indicated that simple pruning techniques can keep computation time low without increasing the error rate. We also plan to use this algorithm to improve the search for the best recognized word chain within a word graph. This search then would make use of all kinds of knowledge sources including prosodic information.

6. References

1. T. Bub and J. Schwinn. Verbmobil: The Evolution of a Complex Large Speech-to-Speech Translation System. In *Int. Conf. on Spoken Language Processing*, volume 4, pages 1026–1029, Philadelphia, 1996.
2. J. Hirschberg and D. Litman. Empirical Studies on the Disambiguation of Cue Phrases. *Computational Linguistics*, 19(3):501–529, 1993.
3. S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz. Dialogue Acts in Verbmobil. Verbmobil Report 65, 1995.
4. R. Kompe, A. Kießling, H. Niemann, E. Nöth, E.G. Schukat-Talamazzini, A. Zottmann, and A. Batliner. Prosodic Scoring of Word Hypotheses Graphs. In *Proc. European Conf. on Speech Communication and Technology*, volume 2, pages 1333–1336, Madrid, 1995.
5. M. Mast, R. Kompe, S. Harbeck, A. Kießling, H. Niemann, E. Nöth, and V. Warnke. Dialog Act Classification with the Help of Prosody. In *Int. Conf. on Spoken Language Processing*, volume 3, pages 1728–1731, Philadelphia, 1996.
6. M. Mast, E. Maier, and B. Schmitz. Criteria for the Segmentation of Spoken Input into Individual Utterances. Verbmobil Report 97, 1995.
7. N.Reithinger, R. Engel, M. Kipp, and Martin Klesen. Predicting Dialog Acts for a Speech–to–Speech Translation System. 1996.
8. E.G. Schukat-Talamazzini. Stochastic Language Models. In *Electrotechnical and Computer Science Conference*, Portorož, Slovenia, 1995.
9. E.G. Schukat-Talamazzini, F. Gallwitz, S. Harbeck, and V. Warnke. Rational Interpolation of Maximum Likelihood Predictors in Stochastic Language Modeling. In *Proc. European Conf. on Speech Communication and Technology*, page to appear, Rhodes, Greece, 1997.