Summarizing Multilingual Spoken Negotiation Dialogues

Norbert Reithinger and **Michael Kipp** and **Ralf Engel** and **Jan Alexandersson***
DFKI GmbH
D-66123 Saarbrücken, Germany
{bert,kipp,rengel,janal}@dfki.de

Abstract

We present the multilingual summarization functionality for VERBMOBIL, a speech translation system. We reuse resources of the system to create a summary. After content extraction, we interpret the results in the dialog context. A summary generator provides the input to generation. A first evaluation indicates the feasibility of the approach.

1 Introduction

In the last decade, automatic summarization of textual (on-line) material was the main goal of programs like TREC and TIPSTER (see e.g. (Mani and Maybury, 1999)). These projects dealt with the summarization of *written* texts. With the availability of speech-based dialogue systems, it is also possible to produce summaries for spoken dialogue.

Within the speech-to-speech translation system VERBMOBIL (Wahlster, 2000), a system that translates negotiations in the domains of scheduling, travel planning, and hotel reservation between German and Japanese or English, we developed summarization facilities that take knowledge sources already present for translation purposes and use them to generate a summary of the translated dialogue.

The rationale behind the summarization in a translation system is to provide the users with notes about the dialogue in their native language. They can be used, e.g., for insertion in schedules, or to check whether the main points of the conversation were correctly recognized and translated by the system.

Our view on summarization is tightly linked to the underlying task of negotiation where you are interested in those objects that all speakers agreed on. In the course of a dialogue many suggestions are brought forward, some are accepted, others rejected, some just forgotten and never mentioned again. In a word, the information is scattered across the dialogue. For summarization, we try to bundle singular data together to form suggestions while keeping track of explicit and implicit statements of acceptance and rejection. The resulting items are presented in a fixed thematic order.

We start by first giving a rough sketch of all modules involved. Then, we show how we robustly extract a content description of utterances from the speech recognizer's output and build a core representation within the dialogue memory. The dialogue processor extends this data and corrects implausible input. We also show how we can use these representations to produce an abstract summary description that is converted by the German language generation module into a natural language summary. By utilizing the transfer component we are able to produce the summary in any language of the VERBMOBIL system. Finally, a first evaluation is presented.

*This work was funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01IV101K/1.

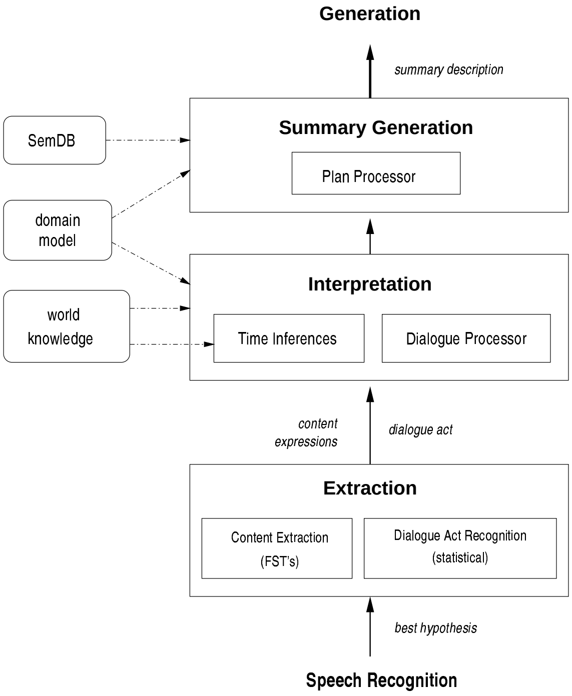

Figure 1: Three-partite architecture for summary generation in VERBMOBIL

2 Architecture

Three VERBMOBIL modules work together to produce a dialogue summary as shown in Figure 1. The input in the processing chain is the best hypothesis from the speech recognizers[1]. The word error rate of the speaker-independent speech recognizers is currently about 20–30%. The best hypothesis is annotated with prosodic information needed for the segmentation of *turns*. A turn is one dialogue contribution and can consists of one or more sentence-like units, henceforth called *utterances*. Since the prosodic information originates from a probabilistic process, the decisions where to insert utterance boundaries are sometimes wrong.

In the *Extraction module* we compute the core intention using a statistical classifier which selects one out of 19 basic dialogue act classes (Reithinger and Klesen, 1997). In parallel, we robustly extract task-relevant content using finite state transducers (Reithinger, 1999). Both, dialogue act and content expression, represent the main content relevant for the domains VERBMOBIL is operating in.

These structures are sent to the *Interpretation module* where they are stored in chronological order as *extracted objects* (ExO's). The module's *dialogue processor* interprets this data in terms of suggestions and attitudes (acceptance, rejection) since our summarization goal is to collect all task-related agreements. To this end, we use discourse and world knowledge to complete the current suggestion with all past data referring to this proposal. The resulting structures are called *negotiation objects* (NeOs), a subset of which – the accepted ones – are then selected as content for the summary.

The *Summary Generation module* is an interface to VERBMOBIL's German natural language generation module. It assembles the thematically structured summary document using interface terms that describe verb, sentence mood, semantic descriptions for events and locations etc. The German syntactic generator of VERBMOBIL produces semantic-syntactic structures for the summary and, in case of an English summary, feeds these structures to the transfer module of VERBMOBIL to obtain the corresponding English structures. After the generation in the target language, the result is marked up with HTML tags for adequate visualization.

3 Extraction

The first step in the processing chain is the extraction of an abstract content representation for each utterance. This functionality was originally developed as a sub-module of a dialogue act based translation module within VERBMOBIL (Reithinger, 1999) and later on emerged as an important part of the dialogue processing chain.

Consider you have to process input like

I would so we were to leave Hamburg on the first
(spoken: good so we will leave Hamburg on the first)

```
[INFORM,has_move:[move,
 has_source_location:[city,has_name='hamburg'],
 has_departure_time:[date,time='day:1']]]
```

[1] VERBMOBIL can be configured with a variety of competing recognizers.

where the recognizer replaced *"good so we will"* with *"I would so we were"*.

The aim is to get an abstract representation of the content and the intention, irrespective of recognition errors like these, as shown under the sentence.

As target representation of the content we use a formalism that comprises the dialogue act which describes the speaker's intention (in the example INFORM) and attribute-value pairs for the content objects (see (Levin et al., 1998) for a comparable approach in speech-to-speech translation systems).

The dialogue act is computed statistically, using language models (Reithinger and Klesen, 1997; Tanaka and Yokoo, 1999). The dialogue act recognizer currently discriminates 19 different types of acts that cover, e.g., suggestions, requests, accepts and rejects, dialogue opening and closing acts, and others. The classifier is trained on a total number of about 1,000 dialogues (consisting of German, English, and Japanese dialogues) which amount to 37,505 utterances. An evaluation where each single dialogue was tested using all the other dialogues as training set resulted in an overall recall value of 72.48% and a precision of 69.90%. The dialogue act is used later in the *dialogue processor* to trigger internal dialogue actions for the summarization process, e.g., SUGGEST adds information, REJECT discards information, utterances marked with GIVE_REASON are ignored.

The second part of the expression describes the extracted content. We have chosen nested attribute-value descriptions for this task. 49 different classes of attribute-value descriptions exist. The extracted information doesn't describe exactly the utterance but is restricted to the propositional content relevant for the summarization process, like locations, dates, hotels, train information, or moving direction (e.g., leaving vs. arriving). The attribute-value descriptions are also specially designed to facilitate the task of combining them in the dialogue processor.

To extract the information we use finite state transducers (FSTs) (Appelt et al., 1993) augmented with functions used, e.g., for scanning input in advance or handling nested objects. The FSTs are hierarchically ordered and grouped in three sequentially processed layers (extracting temporal expressions, creating simple objects using keyword spotting, combining these simple objects into complex ones).

The construction of the FSTs is facilitated by various tools, e.g., a graphical drawing tool for FST development, a syntax checker and several debugging tools. Currently, we have defined 334 multi-language FSTs for the analysis of German, English and Japanese. The FSTs were empirically derived from our sample corpus of about 30,000 utterances.

4 Interpretation

Table 1: The mapping from dialog act to negotiation act and respective processing

dialogue act	*negotiation act*	*processing*
SUGGEST INIT OFFER COMMIT	PROPOSE	(1) complete object (2) compute relation to focussed object (3) focus object
ACCEPT REJECT	FEEDBACK	annotate focussed object with acceptance/rejection
INFORM	ELABORATE	merge object with focussed object
REQUEST	REQUEST	store object in temporary memory

Internally, we model the negotiation in terms of negotiation acts which tell us what objects are part of a suggestion and signal the speakers' attitudes (accept/reject). Suggestions are constantly *completed* (see completion arrow in Fig. 3) and related to previous suggestions by means of the more_specific relation. This allows us to finally select the summary items for generation: the *most specific accepted suggestions*. The whole process is schematically depicted in Fig. 2 and 3 — it will be explained in the rest of this section starting with the introduction of *topics*.

Topic Topics partition our domain into four areas: scheduling, traveling, accommodation

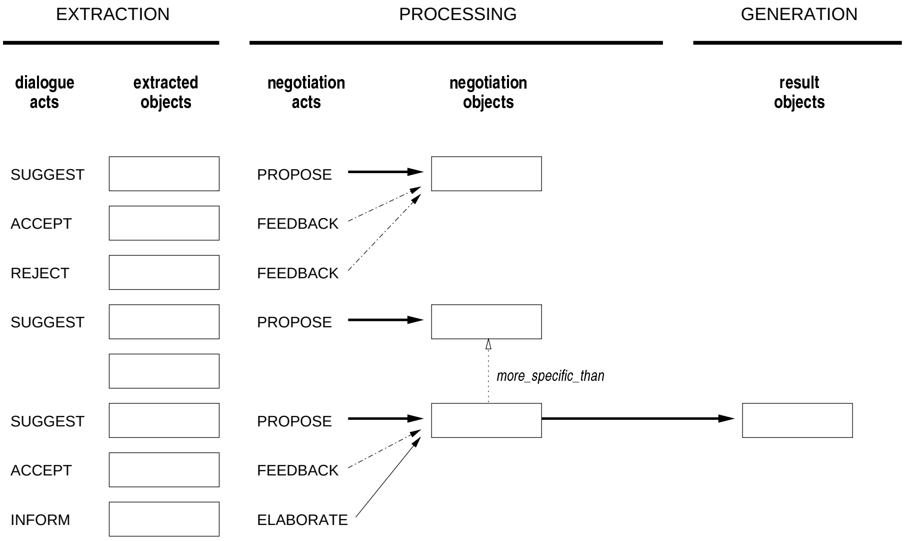

Figure 2: Schematic depiction of the summarization process: dialogue acts are mapped to negotiation acts which control object completion. Attitude annotations (from feedback acts) and inter-object relations (more_specific_than etc.) determine the final selection for the result summary.

and entertainment. To find the topic of an utterance we use keyword spotting plus some heuristics.

Within one topic the speakers are assumed to negotiate a limited set of objects (e.g. objects of the class `journey`, `move` and `book_action` for the traveling topic). We keep a set of templates for each topic where incoming suggestions are integrated to obtain an object we call a *negotiation object* (NeO). In Fig. 3 the original *extracted object* (ExO) of utterance `B2` is integrated in a `journey` template.

For each topic we keep topic-specific information in a *topic frame*. Thus, all suggestions (NeOs) made for one topic are pushed on a topic-specific *focus stack*.

Negotiation acts Whereas the topic serves to insert the ExO into a template to create a NeO, the *negotiation act* determines how to handle the resulting NeO pragmatically. In every negotiation there are essentially four actions that a speaker can perform (1) PROPOSE an object of negotiation, (2) give FEEDBACK on a former proposal, (3) ELABORATE a former proposal by adding matter-of-fact information, or (4) REQUEST task-related information. This information is contained in the dialogue act. Thus, we use a direct mapping to retrieve the negotiation act which in turn controls further processing of the NeO (see Table 1). Negotiation acts can be seen as state transitions in an internal finite state dialogue model. Only these four cases bring about a state change in our dialogue model.[2]

Processing We exemplify the processing of the negotiation acts PROPOSE and FEEDBACK. Consider the dialogue excerpt in Fig. 3 where you see the utterance, ExOs and NeOs. The proposal in `B2` *"there's one at six fourtyfive"* obviously relates to the departure time of the train suggested in `A1` *"let's take the train to Frankfurt"*. Our *completion* process takes care that the NeO of `B2` is expanded to represent the whole implicit suggestion (see section 4 below). At this point we also compute the `more_specific` relation of the new suggestion to all other suggestions made to this point and add the NeO to the topic focus stack. A NeO N_1 is more specific than a previous one N_2 if

- root of N_1 is of the same class or a subclass of N_2

[2] negotiation acts can have parameters as in the case of FEEDBACK where a parameter distinguishes positive (acceptance) and negative (rejection) feedback

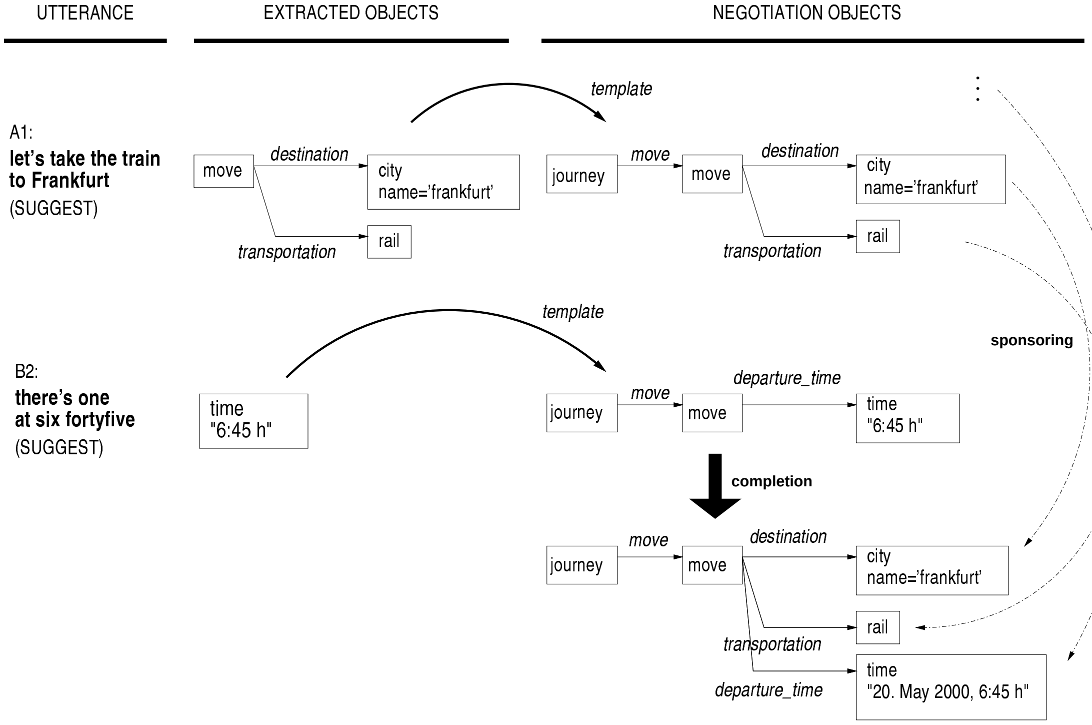

Figure 3: Dialogue excerpt with extracted objects (ExOs) and derived negotiation objects (NeOs).

- for every relation $R(N_2, N')$ there is a relation $R(N_1, N'')$ and N'' is more speciﬁec than or equal to N' where N', N'' are NeO's, temporal objects[3] or primitive data types

FEEDBACK utterances like "alright", "good", "no", "that doesn't work" etc. make the dialogue processor add a respective acceptance/rejection mark to the top NeO on the focus stack.

A schematic depiction of a possible dialogue is shown in Fig. 2. There you can see how attitude annotation and inter-object relations are used to select summary items. We take all objects marked with at least one accept and no reject attitude, and ignore all objects that are related to a `more specific` item. The summary items are passed to summary generation (see Sec. 5).

Completion The completion algorithm's objective is to add information to the current NeO N_{new} from one previous NeO, the so-called *sponsor*. The algorithm consists of (1) finding a suitable sponsoring NeO in the current topic's focus list and (2) taking over parts of the sponsor (see Fig. 3). Both steps are modeled by a single function `complete`(N_{new},N') which tries to complete N_{new} using NeO N' as a sponsor, returning a boolean value for success or failure and leaving N_{new} unchanged in case of failure. By applying this function on every N' on the focus stack until it succeeds[4] we find a sponsor and complete N_{new}.

The function `complete` works recursively through the N_{new} object (and respective subobjects of N'). It first checks certain preconditions: named entities (cities, persons etc.) can only be sponsored by objects with equivalent name, move objects must have certain temporal properties (move back *after* move there) and so on. If the preconditions hold all subtrees of N' that do not occur in N_{new} are added to N_{new} (see Fig. 3). Un-

[3] the relation `more_specific` for temporal objects is equivalent to Allen's relation "in" (see (Allen, 1983))

[4] We found it useful to introduce an upper bound for the number of objects being tested by e.g. 3 (*recency threshold*).

der certain conditions relations can be specialized (e.g. `has_time` to `has_departure_time`). Note that since N' is already a completed object, we obtain a complete object N_{new} without further processing of other preceding objects.

Time expressions are completed by a separate submodule (Kipp et al., 1999).

5 Summary Generation

Responsible for the actual generation of the summaries is the last processing block in Fig. 1 – the summary generator (Alexandersson et al., 2000). On user request it converts the most specific accepted NeOs into sequences of high level German sentence descriptions. These are converted into semantic descriptions (VITs) and finally realized as written text by the existing German generator for presentation. For the generation of, e.g., English summaries, the VITs are sent through the transfer component before realizing them in the English generator.

We characterize the summary planning as simplified text and sentence planning. The summary generator uses an instance of the plan processor described in (Alexandersson and Reithinger, 1997) – for comparable approaches see (Moore, 1989) – which interprets plan operators for traversing the NeOs and partition/convert their content into abstract sentence descriptions.

The information in Verbmobil's semantic database (semdb) has been extended with information about arguments and argument types of the semantic entities for the planning process. For the verbs, optionality of arguments and adjuncts has been added. Verbs, NPs and PPs are basic building blocks for the sentences. The plan processor converts the NeOs, depending on the number of relations and the depth of the content of the relations, to one of the basic building blocks NP, PP and (sequences of) sentences. For simple NeOs (e.g. transportation devices, time expressions) a NP/PP, and for complex NeOs (e.g. `move`, `appointment`) sequences of sentences are generated.

To demonstrate the generation in more detail, consider Fig. 4 which is a NeO that results from a continuation of the dialogue excerpt shown in Fig. 3. Depending on topic (traveling), class (journey) and the content of the top object we select a set of possible verbs. For each verb we recursively generate the content of their appropriate relations yielding a set of NPs, PPs and, eventually, sentences. According to the constraints of the verb (valence roles, sortal constraints for arguments and adjunct(s)) we try to link the NP/PPs to the verb. For *beginnen* the compulsory argument – subject – has to carry the sort *situation*. In this case, we use the move which is related to as `move_there`. This relation corresponds to *Hinreise* (Eng: trip there) which is of sort `move_sit`. *beginnen* also allows for one adjuncts of sort *time_point* and that the source and target location can be linked to the subject. During this process we maintain a context, consisting of, e.g., focus and history list, (cf. (Dale, 1995)) supporting the generation of, e.g., pronouns and demonstratives.

```
Theme: Appointment schedule with trip and
       accommodation

Scheduling: Speaker B and speaker A will
meet in the train station on the 1. of
march 2000 at a quarter to 10 in the
morning.

Traveling: The trip there from Hamburg to
Hanover by train will start on the 2. of
march at 10 o'clock in the morning. The
way back by train will start on the 2. of
march at half past 6 in the evening.

Accommodation: The hotel Luisenhof in
Hanover was agreed on. Speaker A is taking
care of the hotel reservation.
```

Figure 5: An English dialogue summary

This process is iterated until all NeOs are processed. To be robust, we finally use the verb *vereinbaren* (Eng: agree) to realize the relations which were not consumed. The resulting sentence descriptions are passed to the natural language generator which produces the surface structure and provides an HTML-formatted document (Fig. 5).

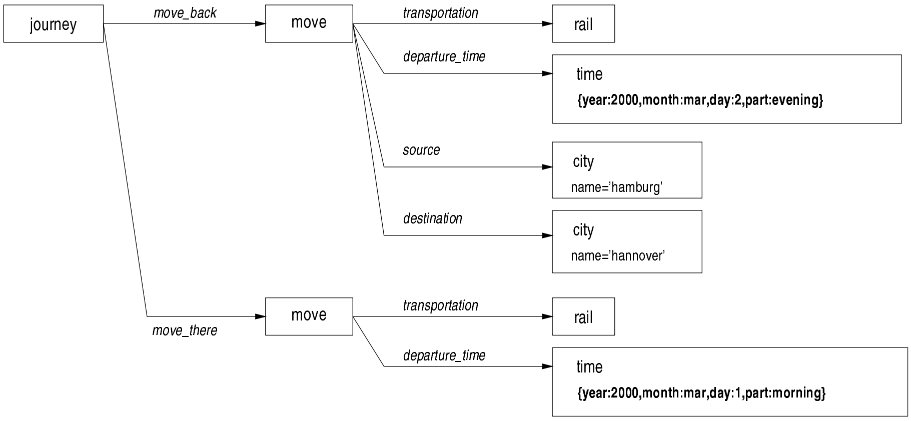

Figure 4: Completed NeO

6 Evaluation

Evaluation is problematic in general, since it is hard to find the ideal summary (Mani and Maybury, 1999). In our case, things are further complicated by the nature of speech-to-speech translation. There are a lot of system errors that can lead to a possible partial breakdown of the dialogue and subsequent repair dialogues. Using, for instance, the recognized and translated utterances as a basis it is in many cases almost impossible, even for a human, to judge what has actual been agreed upon. Consider the excerpt from one of our German-English evaluation dialogues in Fig. 6, where for both participating speakers and an observer it is difficult to grasp what is going on in the dialogue.

T: okay then, that sounds fine. midday at twelve?
H: and sounds fine meet at twelve then
S: ja gut dann lass uns doch um zwölf Uhr treffen
Q: *OK*

T: aber an welchem Tag?
H: aber an welchem Tag
S: but that day suits me as well
Q: *fail; transcripts' translation: "But on which day"*

T: which day did you want to leave? at midday.
H: weeks tight did you want to me at night out
S: wollten Sie für mich erkennen
Q: *fail; translation of system's output: "Would you recognize for me"*

Figure 6: Excerpt from one of our evaluation dialogues. Each block shows **T**ranscription, **H**ypothesis, **S**ystems' Translation, and Translation **Q**uality

Therefore, for a first evaluation we assumed perfect recognition[5] as a starting point and evaluated four German-English dialogues which were mediated and translated by the VERBMOBIL system. During the recording of the dialogues the locutors had no visual contact.

For each of the transcribed dialogues, a human marked the agreed on features, maximally 47 (e.g. location, date for a meeting, speakers name and title, book agent). Each dialogue only contains a subset of these features. The dialogues were run through the system, and the summary was generated. The features in the summary were compared using standard classifications as described in (Mani and Maybury, 1999):

Corr The Feature approximately corresponds to the human annotation. This means that the feature is either (1) a 100% match, (2) it was not enough specified or (3) too specific[6].

Miss A feature is not included.

False A feature was erroneously included in the summary, meaning that the feature was not part of the dialogue or it received a wrong value.

TN (True Negative) A feature was not part of the dialogue, and not included in the summary.

[5] Even using this strategy, it is, as a human, sometimes hard to understand the result of the negotiation.

[6] Example of (2) is when the correct date included a time, which was not captured. Example of (3) is when a date with time was annotated but the feature contained just a date.

Dialogue	1	2	3	4	aver
Turns	33	33	31	32	32.25
Corr	6	13	9	11	9.75
Miss	6	3	5	4	4.5
False	3	3	3	0	2.25
TN	32	28	30	32	30.5
Recall	0.50	0.81	0.64	0.73	0.67
Prec.	0.67	0.81	0.75	1.0	0.81
Fallout	0.09	0.10	0.09	0.00	0.07

Figure 7: Evaluation Results

The results are shown in Fig. 7. As can be seen our approach tries to be on the safe side; the summary contains only those features that the system thinks both partners agreed on. The main reasons for not getting higher numbers is due to the limited recognition of dialogue acts (70% recall) and errors in the content extraction.

7 Conclusion

We demonstrated how one can achieve a summarization functionality of VERBMOBIL by mostly utilizing and extending already existing components. This functionality is fully integrated in the final version of the system.

We use standard methods from the area of natural language processing and information extraction for summarization: Statistical methods are used to compute the intention of an utterance and finite state technology to extract the domain relevant information. The dialogue processor interprets and maintains structures that mirror the negotiated objects and their acceptance status. The summary generator structures the finally agreed on objects partly according to the imposed topic structure and divides the information within each topic to abstract sentence descriptions. These are verbalized and presented by VERBMOBIL's natural language generator. By using the transfer module we can produce multilingual summaries. A first evaluation on a small number of dialogue shows acceptable results for the content contained in the summaries.

Finally, we consider scalability and how to adapt to new domains/tasks and applications: If an already implemented domain is extended, the algorithms can easily be adapted. For new tasks (other than negotiation) the discourse interpretation functionality must be rebuilt. Also, for extending from two speakers to multi-party discussions, a thorough re-structuring of the interpretation is required. In all cases, a corpus of dialogues must be available to be annotated for training and test purposes.

References

J. Alexandersson and N. Reithinger. 1997. Learning Dialogue Structures from a Corpus. In *Proc. of EuroSpeech-97*, pp. 2231–2235, Rhodes.

J. Alexandersson, P. Poller, M. Kipp, and R. Engel. 2000. Multilingual Summary Generation in a Speech–To–Speech Translation System for Multilingual Dialogues. In *Proc. of INLG-2000*, Mitzpe Ramon, Israel.

J. F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, November.

D. E. Appelt, J. Hobbs, J. Bear, and M. Tyson. 1993. FASTUS: A finite-state processor for information extraction from real-world text. In *Proc. of IJCAI-93.*

R. Dale. 1995. An Introduction to Natural Language Generation. Tech. Report, Macquarie University. Presented at ESSLLI-95.

M. Kipp, J. Alexandersson, and N. Reithinger. 1999. Understanding Spontaneous Negotiation Dialogue. In *Workshop Proc. 'Knowledge And Reasoning in Practical Dialogue Systems' of IJCAI '99*, pages 57–64.

L. Levin, D. Gates, A. Lavie, and A. Waibel. 1998. An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues. In *Proc. of ICSLP'98.*

I. Mani and M. Maybury, eds. 1999. *Advances in Automatic Text Summarization.* MIT Press.

J. D. Moore. 1989. *A Reactive Approach to Explanation in Expert and Advice-Giving Systems.* Ph.D. thesis, University of California, L.A.

N. Reithinger and M. Klesen. 1997. Dialogue Act Classification Using Language Models. In *Proc. of EuroSpeech-97*, pages 2235–2238, Rhodes.

N. Reithinger. 1999. Robust Information Extraction in a Speech Translation System. In *Proc. of EuroSpeech-99*, pages 2427–2430.

H. Tanaka and A. Yokoo. 1999. An Efficient Statistical Speech Act Type Tagging System for a Speech Translation System. In *Proc. of ACL-99*, pages 381–388, Baltimore.

W. Wahlster, ed. 2000. *VERBMOBIL: Foundations of Speech-to-Speech Translation.* Springer.