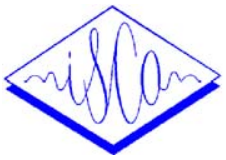
ROBUST INFORMATION EXTRACTION IN A SPEECH TRANSLATION SYSTEM



Norbert Reithinger *

DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken, Germany
E-Mail: norbert.reithinger@dfki.de

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

10.21437/Eurospeech.1999-533

ABSTRACT

Speech processing systems must be able to cope with recognition errors or non grammatical input from the users. We present an approach used in VERBMOBIL to robustly process domain relevant information using cascaded automata. One set of automata is used to extract expressions representing relevant data for the travel planning and hotel reservation domains. This information is e.g. used by the dialogue processing module. Another set of automata is used to generate natural language expressions from these representations. This robust reductionistic translation is one of four translation tracks within VERBMOBIL.

1. INTRODUCTION

Speech processing systems like timetable inquiry systems or speech-to-speech translation systems like VERBMOBIL [2] have to cope with the fact that the word recognition modules do not work with a 100% accuracy. Also the user's spoken language differs sometimes from the ideal of language use.

Therefore, it is often difficult to analyse the output of the speech recognizer by traditional linguistic means. In VERBMOBIL where the domains are travel planning and hotel reservation, this led to the development of an information extraction facility, as in other systems like JANUS [6].

The methods described in this article are currently used by two modules. In order to follow the course of the dialogue the dialogue processing module needs an abstract representation of the main contents of an utterance. A robust recognition of this content is an absolute prerequisite for successfully building up contextual knowledge.

The other module is one of the parallel translation tracks in VERBMOBIL that use different approaches, e.g. statistic and example based translation. The most knowledge intensive is a deep processing track that employs the full power of a thorough linguistic analysis. The approach presented here is implemented in the dialogue act based translation module that provides a robust shallow translation. An abstract representation of the contents from the source language is used to generate a shallow translation in the target language.

The input languages at the time are German, English, and Japanese, whereas the languages generated in the shallow translation module are German and English, with Japanese being under development.

In the rest of the paper we first present the normalized content representation we use and then describe the automaton system and its use for information extraction and language generation.

2. THE CONTENT REPRESENTATIONS

For human-computer or computer mediated human-human interaction systems like VERBMOBIL that work in a limited domain, it is most important to extract the domain relevant information. In the case of VERBMOBIL, the central information is about times and dates, travel directions, hotel reservation and related topics.

To be able to describe and exchange this information, two different formal languages were developed in VERBMOBIL: TEL [5], which formalizes time expressions, and a discourse representation language for travel related information called DRL[1] that is still under development. Both formalisms were developed based on the phenomena found in the VERBMOBIL corpus. This ensures that most of the utterances to be expected in real dialogues can be processed.

Time expressions in TEL abstract from the particular surface use. For example, the German

sechzehn Uhr (*four o'clock pm*)

is translated to

```
[from:[tod:4:0,pod:pm]]
```

*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. The responsibility for the contents of this study lies with the author. Thanks to Ralf Engel and Christian Pietsch for the implementation and definition of the automata, and Ralf Engel, Michael Kipp, and Jan Alexandersson for comments on earlier versions.

[1] [Koch 1998, unpublished technical memo]. We use a "flattened" version derived from the original DRL definition.

where `tod` stands for *t*ime *o*f *d*ay, and `pod` analogously for *p*art *o*f *d*ay. `from` denotes a point label on the time axis and is the default inserted here. TEL also allows more complicated expressions like

from ten to twelve
`[interval:min_between([tod:10:0],[tod:12:0])]`

to express e.g. intervals.

The DRL expressions represent the travel related information like source and destination cities, important hotel related data, and meeting points. An example is

we take the train at seven to Berlin

which is represented as

```
[suggest,traveling,has_move:[move,
 has_date:[date,tempex='tempex(i1,[from:tod:7:0])'],
 has_dest_location:[geo_location,has_name='berlin'],
 has_transportation:[rail]]]
```

This representation contains the dialogue act `suggest` that is computed statistically [7], the dialogue topic `traveling`, the information about the `move`, and the important information that can be extracted related to this move.

The concepts that are dealt with are obviously domain dependent like `move`, `book_action`, `duration` and `date`. They have roles like `has_move`, `has_location`, and `has_book_theme`.

3. THE CASCADED AUTOMATA

For our purposes, namely for providing the dialogue module with propositional information and to produce a shallow translation, it is necessary to employ a very robust and fast extraction system. Initially, we used a definite clause grammar implemented in PROLOG. As the system required more and more functionality, this approach became unmanageable and a major redesign was required.

From the past experience we had the following design requirements

- speed
- robustness
- a user interface to develop knowledge sources

We selected cascaded automata (or transducers) [4] which have a proven record of being successful in various language processing tasks, e.g. for some of the MUC systems [3].

The knowledge source describing the actions on the input string consists of a sequence of folders – and possibly sub-folders – that contain the definition of automata. As usual, an automaton has a start and an end state. The edges between two states are annotated with input symbols that are consumed from the input string and the symbols that are written on the output string. Additionally, the automaton language allows for special condition and action functions e.g. explicitly calling other automata,

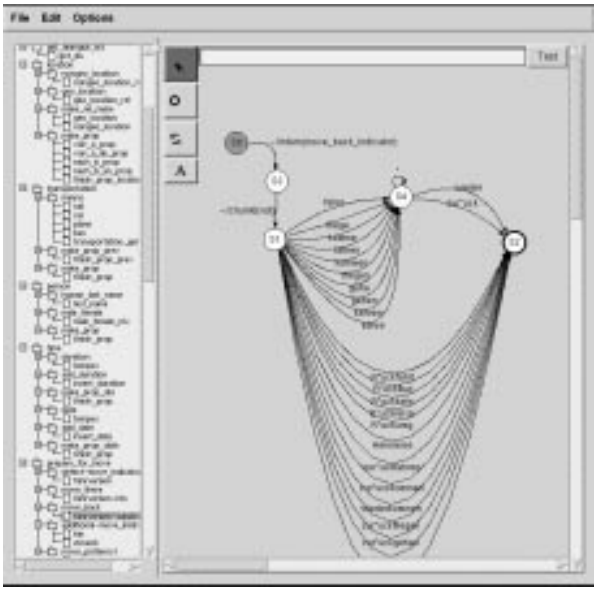


Figure 1: The graphical user interface for automaton development

grouping sequences of symbols together in larger structures and writing this structure back on the output string, checking for the existence of a symbol in the input string, or simple arithmetic functions.

The automata are processed in the sequence as they are defined in the folders. Each of the automata in one folder is tested for applicability until one automaton can be applied successfully or until all automata in the folder are tested. All unprocessed input symbols are then copied to the output string. This proceeds until all folders and the automata contained within are processed.

Since there is no backtracking, the processing is fast and efficient, even with a large knowledge source. Also, since the approach does not require a spanning analysis, it is robust against word recognition errors or words outside the domain of the automata.

To ease the definition and debugging of the system, we developed a user interface that allows the definition of the automata by "drawing" them. Figure 1 shows the surface of the user interface. On the left side the hierarchy of the folders is presented. The major part contains a canvas where the user can draw and change the definition of the automaton.

The interface is implemented in TCL/TK while the automaton processing program is written in C++.

4. USING THE AUTOMATA FOR INFORMATION EXTRACTION AND GENERATION

As stated in the introduction, the information extraction tool is currently used for two tasks, namely to extract information important for the dialogue module and to serve as a basis for dialogue act based translation. Since the two approaches rely on shallow information extraction, it was

considered as highly convenient to reuse programs and knowledge sources as much as possible for both tasks.

We decided to build one set of automata that take as input the best hypothesis from the speech recognizers in the source language and generate the normalized content representation. This representation is used by another set of automata to generate natural language utterances.

4.1. The extraction side

Both parts of the representation, i.e. the TEL and the DRL expressions were developed independently. This allows us to develop the time extraction automata and the DRL extraction as two different automaton hierarchies, too. To get the final automaton definition they are simply pasted together resulting in a unified automaton hierarchy.

The extraction of TEL expressions is rather straightforward. The automata in the first folders of the folder sequence extract idiomatic names for holidays, the next one try to find expressions with numbers like clock times, months and years and finally another layer embeds these expressions in even more complex constructs describing fuzziness and relations between dates.

Complex expressions like

> the twenty ninth thirtieth and thirty first
> `[from:set([dom:29,dom:30,dom:31])]`

are therefore extracted compositionally, first the *d*ay *o*f *m*onths, and then the `set` of all three dates.

For the DRL expressions the problem is that the lexical items expressing them are not as (relatively) narrowly definable as it is the case with temporal expressions which comprise – in their major parts – of only a limited amount of word forms. Even a fancy user interface doesn't free the developer from the tedious task to define the automata to e.g. detect `moves`. Luckily, in VERBMOBIL databases are available that contain semantic knowledge with the distinctions that are needed to identify `moves`, `locations` and other concepts.

From these databases, it is possible to extract automata definitions rather straightforwardly using e.g. PERL scripts. E.g. the automaton that selects the main indicators for a `move` looks like

```
- S0 S1 intern(move_indicator)
travel S1 S2 -
reach S1 S2 -
move S1 S2 -
motion S1 S2 -
...
```

where the first item in the line is the token to be consumed, `S1` and `S2` describe the start and end states of the transition, and – tells the automaton to do nothing. The `intern(move_indicator)` operator writes a non-visible, `intern`al structure named `move_indicator` on the output string that will be later be processed by an automaton down the cascade.

Since the semantic classes in the semantic knowledge sources are the same across all three languages processed currently, it is easy to apply this method for all languages in VERBMOBIL. However, the semantic knowledge sources contain only the stem form and we do not have a morphological analysis. For inflectional languages like German some work by hand is still to be done to insert these forms.

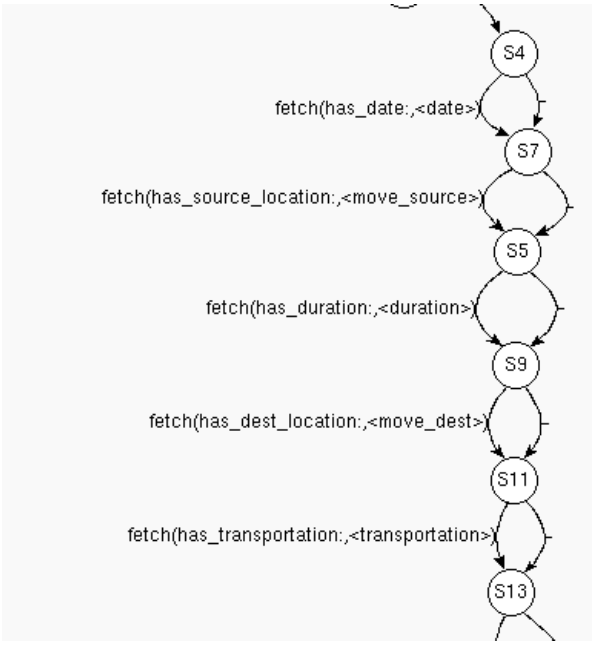


Figure 2: An example of a structure building automaton

The folder and automata sequence for the extraction of the DRL expressions is parallel to that defined for the TEL expressions: first more simple items are searched for in the input, like the abovementioned `move_indicator`. Then this information is grouped together for more complex constructs.

To ease the task of this grouping, we added a special condition that checks for recognized tokens. Figure 2 shows an automaton that builds the structure for the `move` content expression of the example in section 2. The `fetch` function fetches the structure bracketed with $< \cdots >$, and puts the first argument on the output string, followed by the fetched entity.

Parallel to the extraction of the content, we also try to find the topic of the utterance. Topics handled are currently `scheduling`, `accommodation`, `entertainment` and `traveling`. The topic recognition is triggered by keywords and phrases in the input. Finally, the dialogue act that is computed by a different component [7] is added. This expression is then passed on to the dialogue module as a description of the utterance's content.

Currently, the extraction knowledge base consist of 186 automata for German, 167 for English, and 127 for Japanese. For Japanese, only the time expressions are extracted.

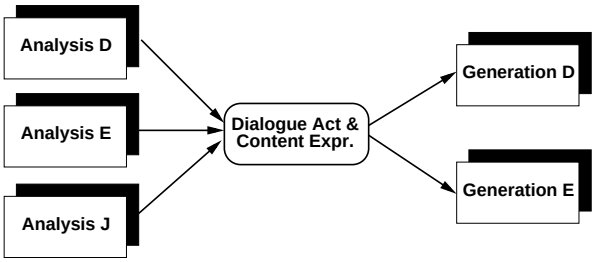


Figure 3: Content expressions as intermediate representation for translation

4.2. The generation side

The second task to be solved with the automata is the generation from the content expressions. We call this approach dialogue act based or shallow translation and it was quite successful in its earlier incarnations in the VERBMOBIL system [2]. The basic approach is to compute the dialogue act statistically [7] for the source language expression, select a suitable sentence pattern for the act in the target language, and splice the translated main topics in this sentence pattern. This is reasonably robust and contributes to the overall coverage of the whole system.

While earlier approaches used a direct translation between language pairs, we now use the content expressions as a sort of very restricted interlingua (see fig. 3). This allows for easy adaption to new languages.

Cascaded automata generate the target language utterances from the content expressions. The division between the generation of the time expressions and of the other expressions is retained for the sake of modularity. The sequence of transformations from the analysis is mirrored in the generation automata: the simple expressions are generated first and are subsequently embedded in more complicated expressions.

One problem remains: find fitting generation patterns for the dialogue acts that on the one hand transport the basic meaning and on the other hand allow for the simple insertion of the propositional information. This is especially problematic in an inflectional language like German. By using fixed prepositions in the verbalization of the dates and roles that determine the inflection we found a tolerable solution.

For the example expression in section 2, the automata generate the following two utterances

> Fahren wir mit dem Zug um sieben Uhr nach Berlin
> How about by train at seven to Berlin

Currently, we use 229 automata to generate German and 195 to generate English.

5. CONCLUSION

We gave a short overview of the current use of cascaded automata to extract and verbalize information in VERBMOBIL. The advantage of this approach can be summarized as follows:

- since no complex analysis is performed recognition errors are tolerated
- if the words related to domain relevant information are recognized the extraction captures the main content
- the definition of the automata is reasonably fast
- a user interface supports the development of the automata

The approach described is integrated in the VERBMOBIL system and successfully provides content information for the dialogue module. For example, the information provided with this system is the main basis for summary generation in VERBMOBIL[1]. The dialogue act based translation realized with this approach is also contributing, as one translation track amongst others, to the overall coverage of the VERBMOBIL system, especially in cases where the recognition is problematic.

Currently, we are fine-tuning both the analysis and generation automata to get the best results of this approach. Also, the Japanese generation will be available soon.

6. REFERENCES

1. Jan Alexandersson and Peter Poller. Towards multilingual protocol generation for spontaneous speech dialogues. In *Proceedings of INLG-98*, Niagara-On-The-Lake, 1998.
2. Thomas Bub, Wolfgang Wahlster, and Alex Waibel. Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proceedings of ICASSP-97*, pages 71–74, Munich, 1997.
3. Nancy Chinchor, editor. *Proceedings of the 7th DARPA Message Understanding Conference*, 1998. Available from `http://www.muc.saic.com/-proceedings/muc_7_toc.html`.
4. David Israel Douglas E. Appelt, Jerry R. Hobbs, John Bear and Mabry Tyson. FASTUS: A finite-state processor for information extraction from real-world text. In *IJCAI-93*, 1993.
5. Ulrich Endriss. Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen. Verbmobil-Report 227, Technische Universität Berlin, 1998. The report is available from the Verbmobil document server at `http://www.dfki.de/cgi-bin/verbmobil/-htbin/doc-access.cgi`.
6. Alon Lavie, Donna Gates, Marsala Gavalda, Laura Mayfield, Alex Waibel, and Lori Levin. Multi-lingual translation of spontaneously spoken language in a limited domain. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 96)*, Copenhagen, 1996.
7. Norbert Reithinger and Martin Klesen. Dialogue act classification using language models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, Rhodes, 1997.