

Speech-to-Speech Translation: The Project Verbmobil

Walther v.Hahn
Jan W. Amtrup

University of Hamburg
Computer Science Department
Natural Language Systems Division
Vogt-Kölln-Str. 30
D-22527 Hamburg

email: {vhahn,amtrup}@informatik.uni-hamburg.de

1 Summary

The research project Verbmobil combines the research branches of continuous speech recognition and machine translation. A consortium of more than 20 universities and 8 companies in Germany as well as two partner institutes in USA and Japan is engaged in the development of a system for translation of spontaneous and continuous speech input to speech output. In the first phase the language pair German and Japanese has been chosen with English as a presentation language (to check the translation quality). The domain of discourse is appointment scheduling. The final application may be an appointment negotiation among two industrial managers from Japan and Germany. In the second project phase the domain will be extended to include travel planning discussions. There exists a prototype implementation integrating the modules of the partners. Additional background research has been carried out to investigate problems e.g. of dialogue and discourse modelling, stochastic parsing and innovative software architectures. This paper gives an overview of the project, its goals and results. It describes in more details research done on integrative architectures. This paper (except section 4) is mostly based on official Verbmobil documents (see Wahlster (1993) and Kay, Gawron, and Norvig (1991)).

2 Aims of the project

Verbmobil combines research activities in continuous speech recognition and in machine translation. It tries to develop a speech-to-speech translation prototype as a feasibility study. The methods to achieve this goal vary among the partners and result in rather heterogeneous modules. Especially in syntax, rule based approaches (mostly “deep” analysis) vs. stochastic approaches (mostly “shallow” understanding components) are tested and refined in parallel. This fact reflects not only the current situation in computational linguistics but may also stem from different cognitive theories about holistic understanding. To handle such heterogeneous components a lot of research has been done on innovative architectures

The technical aim at the end of the first project phase is:

70 % approximately correct translation (end-to-end) on unknown examples within the domain of appointment scheduling and within the lexical set of the given 2285 words.

3 Scientific Background

Spontaneously spoken language has a lot of interesting features most of which make robust processing a rather difficult matter.

- 1 A: Reise(??...??) Grüß Gott
- 2 B: Grüß Gott, ich möchte morgen nach Ulm fahren so gegen 14 Uhr, könnten Sie mir da
- 3 A: gegen bitte?
- 4 B: gegen 14 Uhr möchte ich nach Ulm fahren
- 5 A: gegen 14 Uhr ab Nürnberg um 13 Uhr 58
- 6 B: Ja
- 7 A: dann sind sie in Donauwörth 14 Uhr 53
- 8 B: Ja
- 9 A: gehts weiter 15 Uhr 9
- 10 B: 15 Uhr 9 ab Donauwörth ja
- 11 A: Ja in Ulm 15 Uhr 57
- 12 B: 15 57 ... und einen Zug später?

etc.

(Material by E.Nöth, Erlangen)

You can easily figure out the quality of the corresponding speech material concerning all sorts of omissions, overlapping contributions and linguistic ambiguities, such as:

(Line 7) A: [nsinsin Donauwörth ...]

It is hopeless to look in a traditional way only for single words, or for any subunit of it if you don't have any higher level information about what is

- the dialogue about,
- the state of the dialogue,
- a likely syntactic structure etc.

In other words, in the long run we cannot start as usual from the signal level and pass a set of (word-?) hypotheses to a morphological or syntactic module (and so on) until we obtain, layer by layer, a logical form and a pragmatic interpretation.

For a correct interpretation of realistic natural language communication we might need, e.g., all the suprasegmental, i.e. prosodic information from the signal to help the parser to find structures, to control the pragmatics, to identify speech acts or dialogue moves etc. These contours cannot be understood by the directly adjacent morphology. It is not reasonable to accumulate information and transport it stepwise until it reaches the component where the information is needed and processed. It should be sent directly to the module that needs it. More arguments for top down information (supporting decisions on lower levels) as well as bottom-up communication (not even between adjacent layers) are given in section 4.

However, nobody yet has the necessary tools and a systematic testbed (with enough complexity and generality) to scrutinise systematically what is the optimal interaction in terms of

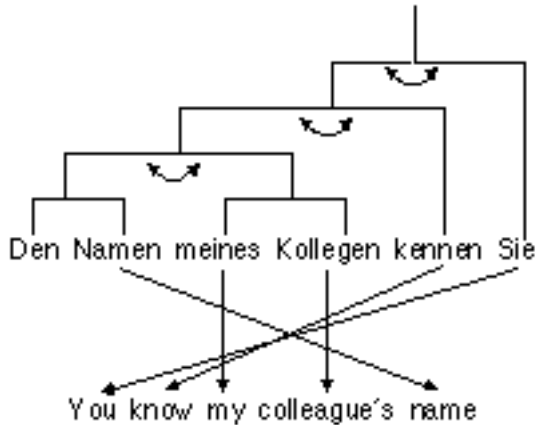
- features of links between modules,
- amount of interaction,
- time behaviour of modules,
- reduction of ambiguity
- its linguistic correlates,
- the achieved reduction in search, resources and time, or even
- cognitive hypotheses about language understanding.

Therefore we investigate systematically these effects in a completely declarative and transparent environment. Our results are used as the basis for an optimised system with hybrid components applying completely different techniques, formalisms and ways of interaction (as explained in more detail in section 4).

Let us now address some difficulties of (technical) translation, as Kay showed in Kay (1996). Different languages exhibit completely different structures of terminology:

remove the spark plugs	Zündkerzen herausdrehen
remove the plug leads	Zündkabel abziehen
remove the dipstick	Ölmeßstab herausziehen
remove filter cap	Verschlußkappe aufdrehen
remove distributor cap	Verteilerdeckel abnehmen
remove rotor arm	Verteilerläufer abziehen
remove nipple	Schmiernippel herausdrehen
remove the two bolts	Beide Schrauben lösen

They differ widely in word order:



Or they have features unknown in one of the languages, e.g., honorifics in Japanese, but not in German.

A combination of architectural problems and translation strategies occurs, when an utterance which is highly redundant, contains äms, hms, or other noise, which has syntactic or lexical repairs or is structurally “incorrect” should be translated to a compact and clear sentence: What does compressing translation technically mean to the single components? How many variations are personal style or intended ambiguity?

4 Flexible Architectures

Verbmobil in itself is a very large, distributed system. It offers many configuration options that have a direct affect to the system architecture. This regards especially the integration of modules of different origin for identical purposes. E.g., there are currently two distinct word recognizers in use, three versions of the syntactic/semantic

processing exist. Additionally, a whole group within the Verbmobil project is concerned with the planning and evaluation of new architectures. In this section, we will elaborate on the communication system used within Verbmobil which offers several paths for setting up a flexible architecture. Furthermore, we will introduce work done by the architectural subproject of Verbmobil. This work focusses on incremental, interactive architectures that may offer future advantages compared to more conservative, almost always non-incremental systems.

4.1 An approach to communication for NLP systems

Complex natural language processing systems that integrate multimodal interfaces with speech or vision are nowadays often built in a distributed, agent-oriented manner. The reasons for this kind of architecture include speed (due to the distribution among several computers) and interface issues which can be solved best by encapsulating functions into modules. Verbmobil consists of more than thirty functional modules, written in six different programming languages. In order to avoid confusion by implementing different communication facilities for at least some of the interfaces between components, we designed and implemented ICE (*Intarc Communication Environment*), a communication subsystem suitable for heterogeneous hardware and software architectures and supporting multiple programming languages (Amtrup, 1995).

The central aims guiding the design of ICE were

- to enable developers of components to encode communication facilities in a simple, elegant way, thus reducing the effort to be spent in interface issues. This allows developers to concentrate on the scientific problems at hand.
- to provide a sound theoretical basis of communication and to integrate this notion into a standard portable message passing system.
- to allow for several configuration options that enable the construction of system variants

and the evaluation of different system characteristics.

We chose to adapt the channel model (Hoare, 1978) which offers a simple, yet well-founded sight onto process communication. The original model was extended to allow asynchronous message passing and the integration of programming-language specific complex data types. The low level routines of our communication system are given by PVM, the *Parallel Virtual Machine* (Geist et al., 1994), a widespread process-communication software for a large variety of hardware architectures.

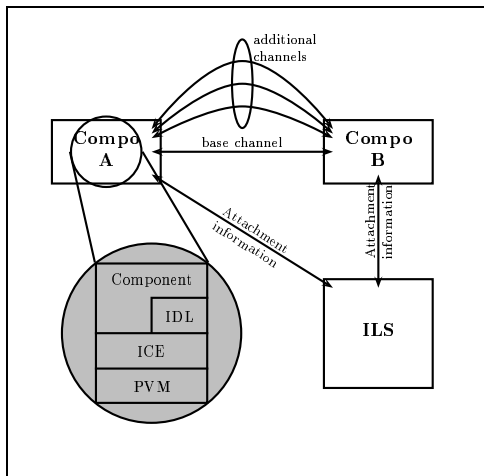


Figure 1: Principle component layout

Figure 1 shows the overall layout of a system using ICE as a communication mechanism. All communication is done by means of channels, which represent bidirectional point-to-point connections. Messages can be sent either way asynchronously which reduces the delay due to component synchronisation included in the original CSP model (Hoare, 1978). We distinguish between *base channels*, which are the primary means of communication among components and *additional channels*, which offer a greater bandwidth of functionality and configuration options. Base channels are configured in a way guaranteeing that each component is able to interact with every other component. This is achieved by using the PVM-standard way of communication. Additional channels can be used to separate different data and control streams or to speed up communication by remov-

ing an additional encoding step for the information exchange between identical computer architectures.

Both types of channels can be configured further. Ordinarily, channels are connections between two points (normally components). We introduced the possibility to split channels both on the sending and receiving sides. By splitting channels it is easy to attach additional listeners to a channel, e.g. for visualisation purposes. Furthermore, the data sent on a channel may be guided through an additional module which performs modifications on the data to take into account differences in interface specifications (e.g. when mixing different versions of modules). The components at the ends of the channel are unaware of the splitting which guarantees identical system behavior even in complex channel configurations.

A component is equipped with three different software layers when operating with ICE:

- The core communication routines are driven by PVM. This includes the actual message passing and routing functions.
- The central ICE layer provides functions for attaching a component to the overall system, for the establishment of channels as well as some information methods. The interfaces for individual programming languages reside here. Currently, we support C, C++, Lisp (four dialects), Prolog (two dialects) and Tcl/Tk.
- When complex data types are to be communicated, the third layer (IDL, *Intarc Data Layer*) comes into play. It provides hooks for encoding and decoding routines for several user-defined data types, including signal data and semantic specifications used within Verbmobil.

In order to accomplish identification and configuration for the whole system, a dedicated component (ILS, *Intarc License Server*) has been introduced. It contains information about the system configuration and all enrolled components and channels. Components interact with the ILS upon

attaching to the system and during each individual channel creation. Afterwards, communication is strictly point-to-point, thus no additional overhead is generated during operation.

4.2 Incremental architectures for Verbmobil

The communication system ICE has not only been used for the various Verbmobil demonstrators and prototypes (Amtrup and Benra, 1996), but is also incorporated into the experimental system architecture (INTARC) developed by a subproject within Verbmobil.¹ The approach chosen for INTARC is to develop a speech translation system obeying design principles that have their origin in the goal of reflecting some of the assumed properties of human speech processing.

Two of the key features of human speech processing are integrated into the system: Incrementality and interactivity. Incrementality denotes the piecemeal fashion in which speech is understood by humans. E.g., word recognition starts as soon as some part of the input signal is available. The assumption made by the *cohort model* of Marslen-Wilson and others (see, e.g., Marslen-Wilson and Tyler (1980)) is that continuously larger intervals of the input signal are mapped onto a set of words in the lexicon matching the amount of speech heard, until finally one word remains active in the cohort which is assumed to be the one actually spoken. This property is not restricted to word recognition, but extends to syntactic/semantic processing (Niv, 1993).

The investigation of context effects within speech processing suggests the second feature important for INTARC. Different levels of linguistic processing are not completely independent of each other, but interact to a certain amount (Zwitserslood, 1989).

Both properties have been integrated into the architecture of INTARC. Each component works strictly from left to right and delivers partial re-

sults as soon as they are computed (i.e., incrementally). The effect of this operation is that several modules are able to work in parallel on almost identical intervals of the speech signal. As soon as the speech recognizer completes word hypotheses for a given frame, they can be sent to a syntactic parser which in turn can start to compute partial utterance hypotheses based on word recognition. Thus, the inherent sequential character of many systems can be overcome, where the first operation is to compute word hypotheses for the whole utterance before any word information is sent to linguistic modules.

By working incrementally, interactivity between different modules is possible. In principle, interaction may be used for two objectives:

- It may lead to a reduction of a search space of a component. Consider the case of a speech recognizer sending word hypotheses to a syntactic parser. The parser is able to decide whether or not a given word fits into the present syntactic derivation. If not, this fact is communicated backwards to the speech recognizer which can stop all processing depending on that particular word hypothesis.
- It may be used to generate hypotheses. To recur to the example just mentioned, a parser can generate a set of words fitting into the derivation at every point in time. This set can be delivered to the speech recognizer. By examining that set, the shape of the search space itself is modified and only promising words are set up to be recognized.

Both models of interaction may lead to a reduction of bandwidth needed between components and thus increase the overall performance. Moreover, more elaborate algorithms may then be used. Both schemata of interaction have been investigated within the INTARC architecture (Hauenstein and Weber, 1994). A tight interaction schedule was introduced that resulted in a synchronization between speech recognizer and syntactic parser every 10 ms.

¹In fact, originally, the primary purpose of ICE was to be used for INTARC (*INT*eractive *ARC*itecture), the experimental system; hence the name ICE.

5 Future Aims

The next project phase will test the robustness and the applicability of the present system and enhance the system in the following directions:

- Higher lexicon coverage, which means increasing the number of entries to more than 3000
- Switching to telephone quality or even free-speaking facility
- Domain switching with automatic detection
- Higher mobility by establishing a translation server used by radio link
- Multilinguality: Including French
- Multimedia: Vision channel
- Multifunctionality: Data base queries, booking tickets...
- Multiparty extension: Several partners converse in several languages
- Dialog minutes on request and paraphrase mode in the source language

6 Bibliography

- Amtrup, Jan W. 1995. ICE-Intarc Communication Environment: User's Guide and Reference Manual. Version 1.4. Verbmobil Technical Document 14, Univ. of Hamburg, December.
- Amtrup, Jan W. and Jörg Benra. 1996. Communication in large distributed AI Systems for Natural Language Processing. In *Proceedings of the 16th international Conference on Computational Linguistics*, pages 35-40, Copenhagen, Denmark, August.
- Geist, Al, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. 1994. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing*. Cambridge, MA: MIT Press.
- Hauenstein, Andreas and Hans Weber. 1994. An Investigation of Tightly Coupled Speech Language Interfaces Using an Unification Grammar. In *Proceedings of the Workshop on Integration of Natural Language and Speech Processing at AAAI '94*, pages 42-50, Seattle, WA.
- Hoare, Charles A. Richard. 1978. Communicating Sequential Processes. *Communications of the ACM*, 21(8):666-677, August.
- Kay, M. 1996. Tutorial: Machine Translation. In C. Burgard, R. Burgard, and R. Karger, editors, *Foliendokumentation der Vorträge zur 4. Projektlenkungssitzung für Verbmobil. Techn. Dokument 43*. Aachen.
- Kay, M., J.M. Gawron, and P. Norvig. 1991. *Verbmobil: A Translation System for Face-to-Face Dialog*. CSLI.
- Marslen-Wilson, William D. and Lorraine K. Tyler. 1980. The temporal structure of spoken language understanding. *Cognition*, 8:1-71.
- Niv, Michael. 1993. *A Computational Model of Syntactic Processing: Ambiguity Resolution from Interpretation*. Ph.D. thesis, Univ. of Pennsylvania.
- Wahlster, Wolfgang. 1993. Translation of Face-to-Face-Dialogs. In *Proc. MT Summit IV*, pages 127-135, Kobe, Japan.
- Zwitserslood, P. 1989. The locus of effects of sentential-semantic context in spoken-word processing. *Cognition*, 32:25-64.