# From Offline Evaluation to Online Selection, (13)

Damir Ćavar, Uwe Küssner, and Dan Tidhar

Technische Universität Berlin, Germany

**Abstract.** In this chapter we describe some of the problems that arise from the need to integrate four alternative translations for each input utterance, and to come up with exactly one optimal translation. In the center of this chapter is a learning method that was tailored to overcome the problem of incomparable confidence values delivered by the four competing translation paths. By using offline human feedback and applying a linear optimization heuristic, we determine a rescaling scheme that enable us to compare confidence values across modules. We also describe some additional information sources that further elaborate the selection procedure, and finally, outline some Quality of Service parameters that are supported by the selection module.

## 1 Introduction

For the language pairs English-German and German-English, Verbmobil applies four different translation methods that operate in parallel, according to four alternative approaches to machine translation, thus increasing the system's robustness and versatility. Since the system should always produce exactly one translation for each input utterance that it encounters, a selection procedure is necessary, which chooses the best alternative for each given utterance. Furthermore, in order to benefit from the diversity of translation methods, the alternative translations are also combined within the boundaries of single utterances, so as to form new compound translations. The selection procedure relies on confidence values that are delivered together with the translated segments from each of the alternative translation components. Since the confidence values are computed by independent components that are based on fundamentally different MT strategies, they are not directly comparable, neither with each other, nor with the objective judgments of human evaluators, i.e. they need to be rescaled in order to gain comparative significance. In the following sections, we describe the strategy we have explored in order to acquire the necessary human annotations of the translation quality, that are used as the bootstrapping data for an optimized confidence rescaling schema. For our purpose, the results from human evaluation are the key data. In order to guarantee maximal reliability we make use of different strategies developed and used for example in experimental psychology. The annotation task itself has to be designed in a way to resolve optimally the tension between the need to be maximally easy for the evaluators

(low time resources, low cognitive effort) and maximally reliable and usable for the developers. The tasks for the evaluator were set up to consist of the following primitives: a. simple reading task, b. binary decision task, c. simple counting task, and the possibility to make notes. As is well known from experimental psychology and psycholinguistics, simple binary decision tasks (e.g. yes/no questions), for example, are answered much faster, and more reliably across items and across subjects by the evaluators than decision tasks that provide a decision scale. Counting tasks are used, both, to generate relevant annotated data (e.g. the number of relevant information units), and to provide automatic means for checking the evaluators reliability, given that some counting tasks can be performed automatically. The evaluator, for example, is randomly asked to count the number of words in either the input or the output, and the instructor explains that this is relevant for the evaluation. The number of mistakes the evaluator makes can be used to relativize the other evaluation results automatically. Such evaluation strategies and precise instructions, as we experienced, give very robust results. Based on this evaluation, a set of 'off line' confidence values is calculated, and a list of alternative segment combinations is produced, sorted according to their corresponding off line confidence values. The annotators then process these lists in a second annotation phase, in which they are requested to select from each list a minimal subset of 'best' translations. The results of this second annotation round are then combined with the original 'on line' confidence values to form inequalities that express the annotators' preferences as a set of constraints on the linear rescaling coefficients.

### 1.1 The Incomparability Problem

Each translation module calculates a confidence value for each of the translations that it produces, to serve as a guiding criterion for the selection procedure. However, since the various translation methods are fundamentally different from one another, the resulting confidence values cannot be compared per se. Whereas we do assume a general correspondence between confidence values and translation quality within each one of the modules, there is no guaranty whatsoever that a high value delivered by a certain module would indeed signify a better translation when compared with another value, even a much lower one, which was delivered by another module. An additional step needs to be taken in order to make the confidence values comparable with one another.

### 1.2 Working Hypotheses

The task of evaluating translation quality is non-trivial also for human annotators, since the applicable criteria are diverse, and at the absence of a comprehensive translation theory, very often lead to contradicting conclusions. This difficulty is partially dealt with below, but for practical reasons we tend to accept the need to rely on human judgment, partially theory assisted and partially intuitive, as inevitable. Another important presupposition underlying the current solution, is that the desirable

rescaling can be well approximated by means of linear polynomials. The computational benefits of this assumption are immense, as it allows us to remain within the relatively friendly realm of linear equations (albeit inconsistent). The price that we have to pay in terms of precision is not as big as one might expect, because the crucial matter to our case is the comparative behavior of the obtained confidence curves, i.e. the breakpoints in which one overtakes the other, rather than the precise details of their behavior in between. Note that from the expectation that confidence values would indeed reflect the translation quality it follows that the rescaling should be monotonous, which makes linear approximation even more appropriate.

### 1.3 The Various Translation Paths

The input shared by all four translation paths consists of sequences of annotated *Word Hypotheses Graphs (WHG)*. Each *WHG* is produced by a speaker independent voice recognition module, and is annotated with additional prosodic information and pause information by a prosody module (Buckow et al., 1998). Each translation subsystem selects independently both a path through the *WHG*, and a possible segmentation, according to its grammar and to the prosody information. This implies that even though all translation paths are sharing the same input data structure, both the chosen input string and its chosen segmentation may well be different for each path. In this section we provide the reader with very brief descriptions of the different translation subsystems, along with their respective methods for calculating confidence values.

- The **ali** subsystem implements an example based translation approach. Confidence values are calculated according to the matching level of the input string with its counterparts in the example database.
- The **stattrans** (Och et al., 1999) subsystem is a statistical translation system. Confidence values are calculated according to a statistical language model of the target language, in conjunction with a statistical translation model.
- The **syndialog** (Kipp et al., 1999) subsystem is a dialogue act based translation system. The translation invariant consists of a recognized dialogue act, together with its extracted propositional content. The confidence value reflects the probability that the dialogue act has been correctly recognized, and the extent to which the propositional content has been successfully extracted.
- The **deep** translation path in itself consists of multiple pipelined modules: linguistic analysis, semantic construction, dialogue and discourse semantics, and transfer (Emele and Dorna, 1996) and generation (Kilger and Finkler, 1995) components. The transfer module is supported by disambiguation information from the context (Koch et al., 2000) and dialogue modules. The linguistic analysis part consists of several parsers which, in turn, also operate in parallel (Ruland et al., 1998). They include an HPSG parser, a chunk parser and a statistical parser, all producing data structures of the same kind, namely, the *Verbmobil Interface Terms (VITs)* (Dorna, 1999). Therefore, within the deep processing path, a selection problem arises, similar to the larger scale problem of selecting

the best translation. The internal selection process within the deep translation path is based on a probabilistic *VIT* model. Confidence values within the deep path are computed according to the amount of coverage of the input string by the selected parse, and are subject to modifications as a byproduct of combining and repairing rules applied by the semantics mechanism. Additional information which influences the 'deep' confidence values is provided by the generation module, which estimates the percentage of each transferred *VIT* which can be successfully realized in the target language.

Although all confidence values are finally scaled to the interval $[0, 100]$ by their respective generating modules, there seems to be hardly any reason to believe that such fundamentally different calculation methods would yield magnitudes that are directly comparable with one another. As expected, our experience has shown that when confidence values are taken as such, without any further modification, their comparative significance is indeed very limited.

## 2 The Selection Procedure

In order to improve their comparative significance, the delivered confidence values $c(s)$, for each given segment $s$, are rescaled by linear functions of the form:

$$a \cdot c(s) + b \ . \tag{1}$$

Note that each input utterance is decomposed into several segments independently, and hence potentially differently, by each of the translation paths. The different segments are then combined to form a data structure which, by analogy to *Word Hypotheses Graph*, can be called *Translation Alternatives Graph (TAG)*. The size of this graph is bound by $4^n$, which is reached if all translation paths happen to choose an identical partition into exactly $n$ segments. The following vector notation was adopted in order to simplify simultaneous reference to all translation paths. The linear coefficients are represented by the following four-dimensional vectors:

$$\mathbf{a} = \begin{pmatrix} a_{ali} \\ a_{syndialog} \\ a_{stattrans} \\ a_{deep} \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_{ali} \\ b_{syndialog} \\ b_{stattrans} \\ b_{deep} \end{pmatrix} \ . \tag{2}$$

Single vector components can then be referred to by simple projections, if we represent the different translation paths as orthogonal unit vectors, so that $\mathbf{s}$ denotes the vector corresponding to the module by which $s$ had been generated. The normalized confidence is then represented by:

$$(\mathbf{a} \cdot k(s) + \mathbf{b}) \cdot \mathbf{s} \ . \tag{3}$$

In order to favor translations with higher input string coverage, the compared magnitudes are actually the (rescaled) confidence values integrated with respect to the

time axis, rather than the (rescaled) confidence values as such. Let $\|s\|$ be the length of a segment $s$ of the input stream, in milliseconds. Let **SEQ** be the set of all possible segment sequences within the TAG, and $Seq \in$ **SEQ** any particular sequence.

We define the normalized confidence of $Seq$ as follows:

$$C(Seq) = \sum_{s \in Seq} ((\mathbf{a} \cdot c(s) + \mathbf{b}) \cdot \mathbf{s}) \cdot \|s\| \ . \tag{4}$$

This induces the following order relation:

$$seq_1 \leq_C seq_2 \stackrel{def}{=} C(seq_1) \leq C(seq_2) \ . \tag{5}$$

Based on this relation, we define the set of best sequences as follows:

$$Best(\mathbf{SEQ}) = \{seq \in \mathbf{SEQ} \mid seq \text{ is a maximum element in } (\mathbf{SEQ}; \leq_C)\} \ . \tag{6}$$

The selection procedure consists in generating the various possible sequences, computing their respective normalized confidence values, and arbitrarily choosing a member of the set of best sequences. It should be noted that not all sequences need to be actually generated and tested, due to the incorporation of Dijkstra's well known "Shortest Path" algorithm (e.g. in Cormen et al., 1989).

## 3 The Learning Cycle

Learning the rescaling coefficients is performed off line, and should normally take place only once, unless new training data is assembled, or new criteria for the desirable system behavior have been formulated. The learning cycle consists of incorporating human feedback (training set annotation) and finding a set of rescaling coefficients so as to yield a selection procedure with optimal or close to optimal accord with the human annotations. The first step in the learning procedure is choosing the set of training data. This choice has a direct influence on the learning's result, and, of course, on the amount of time and resources that it requires. In the course of our work we've performed this procedure several times, with training sets of various sizes, all taken from a corpus of test dialogues, designed to provide a reasonable coverage of the desirable functionality of the current Verbmobil version. Since the optimization algorithm (described below) normally terminates within no more than a couple of hours, the main bottle neck in terms of time consumption have normally been the human annotators. With what appears to be, from our experience, a reasonably large training set, i.e. a set of 7 from the above mentioned test dialogues (including 240 dialogue turns and 1980 different segments), the complete learning cycle can be performed within a few days, depending on the annotators' diligence, of course. Once a training set has been determined, it is first fed through the system, while separately storing the outputs produced by the various translation modules. The system's output is then subject to two phases of annotation, resulting in a uniquely determined 'best' sequence of translated segments for each input

utterance. The next task is to learn the appropriate linear rescaling, that would maximize the accord between the new, rescaled confidence values, and the preferences dictated by the newly given 'best' sequences. In order to do that, we first generate a large set of inequalities, and then obtain their optimal, or close to optimal solution.

### 3.1 Training Set Annotation

The two annotation phases can be described as follows: first, the outputs of the alternative translations paths are annotated separately, so as to enable the calculation of the 'offline confidence values' as described below. For each dialogue turn, all possible combinations of translated segments that cover the input are then generated. For each of those possible combinations, an overall off line confidence value is calculated, in a similar way to which the 'online' confidence is calculated, leaving out the rescaling coefficients, but keeping the time axis integration. These segment combinations are then presented to the annotators for a second round, sorted according to their respective off line confidence values. The annotator is requested at this stage merely to select the best segment combination, which would normally be one of the first to appear on the list. The first annotation stage may be described as 'theory assisted annotation', and the second is its more intuitive complement. To assist the first annotation round we have compiled a set of annotation criteria, and designed a specialized annotation tool for their application. These criteria direct the annotator's attention to 'essential information items', and refer to the number of such items that have been deleted, inserted or maintained during the translation. Other criteria are the semantic and syntactic correctness of the translated utterance as well as those of the source utterance. The separate annotation of these criteria allows us to express the 'offline confidence' as their weighted linear combination. The different weights can be seen as implicitly establishing a method of quantifying translation quality. One can determine, for instance, which is of higher importance — syntactical correctness, or the transmission of all essential information items. Using the vague notion of 'translation quality' as a single criterion would have definitely caused a great divergence in personal annotation style and preferences, as can be very well exemplified by the case of the dialogue act based translation: some people find word by word correctness of a translation much more important than the dialogue act invariance, while others argue exactly the opposite (Schmitz, 1997 and Schmitz and Quantz, 1995). The adequacy of these linear combinations is then verified by comparison to explicit human selection, which can in fact be reasonably well predicted by the resulting offline confidence values.

### 3.2 Generating Inequalities

Once the best segment sequences for each utterance have been determined by the completed annotation procedure, a set of inequalities is created using the linear rescaling coefficients as variables. This is done simply by stating the requirement that the normalized confidence value of the best segment sequence should be better than the normalized confidence values of each one of the other possible sequences.

For each utterance with $n$ possible segment sequences, this requirement is expressed by $(n-1)$ inequalities. It is worth mentioning at this point that it sometimes occurs during the second annotation phase, that numerous sequences relating to the same utterance are considered 'equally best' by the annotator. In such cases, when not *all* sequences are concerned but only a subset of all possible sequences, we have allowed the annotator to select multiple sequences as 'best', correspondingly multiplying the number of inequalities that are introduced by the utterance in question. These multiple sets are known in advance to be inconsistent, as they in fact formulate contradictory requirements. Since the optimization procedure attempts to satisfy the largest possible subset of inequalities, the logical relation between such contradicting sets can be seen as disjunction rather than conjunction, and they do seem to contribute to the learning process, because the different 'equally best' sequences are still favored in comparison to all other sequences relating to the same utterance. The overall resulting set of inequalities is normally very large, and can be expected to be consistent only in a very idealized world, even in the absence of 'equally best' annotations (which are inconsistent by definition). The inconsistencies reflect many imperfections that characterize both the problem at hand and the long way to its solution, most outstanding of which is the fact that the original confidence values, as useful as they may be, are nevertheless far from reflecting the human annotation and evaluation results, which are, furthermore, not always consistent among themselves. The rest of the learning process consists in trying to satisfy as many inequalities as possible without reaching a contradiction.

### 3.3 Optimization Heuristics

The problem of finding the best rescaling coefficients reduces itself, under the above mentioned presuppositions, to that of finding the maximal consistent subset of inequalities within a larger, most likely inconsistent, set of linear inequalities, and solving it. In (Amaldi and Mattavelli, 1997), the problem of extracting close-to-maximum consistent subsystems from an inconsistent linear system (MAX CS) is treated as part of a strategy for solving the problem of partitioning an inconsistent linear system into a minimal number of consistent subsystems (MIN PCS). Both problems are NP-hard, but through a thermal variation of previous work by Agmon (1954) and Motzkin and Schoenberg (1954), a greedy algorithm is formulated by Amaldi and Mattavelli (1997), which can serve as an effective heuristic for obtaining optimal or near to optimal solutions for MAX CS. Implementing this algorithm in the C language enabled us to complete the learning cycle by finding a set of coefficients that maximizes, or at least nearly maximizes, the accord of the rescaled confidence values with the judgment provided by human annotators.

## 4 Additional Information Sources

Independently of the confidence rescaling process, we have made several attempts to incorporate additional knowledge sources in order to refine the selection procedure.

Some of these attempts, such as using probabilistic language model information, or inferring from the logical relation between the approximated propositional contents of neighboring utterances (e.g. trying to eliminate contradiction), have so far not been fruitful enough to be worth full description in the present work. Two other attempts do seem to be worth mentioning in further detail, namely, using dialogue act information, and using disambiguation information, which are described in the following two sections.

### 4.1 Dialogue Act Information

Our experience shows that the translation quality that is accomplished by the different modules varies, among the rest, according to the dialogue act at hand. This seems to be particularly true for **syndialog**, the dialogue act based translation path. Those dialogue acts that normally transmit very little propositional content, or those that transmit no propositional content at all, are normally handled better by **syndialog** compared to dialogue acts that transmit more information (such as INFORM, which can in principle transmit any proposition). The dialogue act recognition algorithm used by **syndialog** does not compute the single most likely dialog act, but rather a probability distribution of all possible dialogue acts[1] We represent the dialogue act probability distribution for a given segment $s$ by the vector $\mathbf{da}(s)$, where each component denotes the conditional probability of a certain dialogue act, given the segment $s$:

$$\mathbf{da}(s) = \begin{pmatrix} P(suggest|s) \\ P(reject|s) \\ P(greet|s) \\ \vdots \end{pmatrix} \quad . \tag{7}$$

The vectors $\mathbf{a}$ and $\mathbf{b}$, that have been introduced for describing the selection procedure, are replaced by the matrices $\mathbf{A}$ and $\mathbf{B}$ which are simply a concatenation of the respective dialogue act vectors.
Let $\mathbf{A}^s = \mathbf{A} \cdot \mathbf{da}(s)$, and $\mathbf{B}^s = \mathbf{B} \cdot \mathbf{da}(s)$.
The normalized confidence value, with incorporated dialogue act information can then be expressed as:

$$C(Seq) = \sum_{s \in Seq} ((\mathbf{A}^s \cdot c(s) + \mathbf{B}^s) \cdot \mathbf{s}) \cdot \|s\| \quad . \tag{8}$$

### 4.2 Disambiguation Information

Within the **deep** translation path, several types of underspecification are used for representing ambiguities (Küssner, 1997, Küssner, 1998, and Emele and Dorna, 1998). Whenever an ambiguity has to be resolved in order for the translation to succeed, resolution is triggered on demand (Buschbeck-Wolf, 1997). Several types of

[1] For more information about dialogue acts in Verbmobil, see Alexandersson et al. (1997)

disambiguation are performed by the context module (Koch et al., 2000). Within this module, several knowledge sources are used in conjunction for resolving anaphora and lexical ambiguities. Examples for such knowledge sources are world knowledge, knowledge about the dialogue state, as well as various sorts of morphological, syntactic and semantic information. Additionally, dialogue act recognition is performed, and a representation of the main dialogue turn content is constructed. Of considerable importance to the Verbmobil scenario are the representations and reasoning on date and time expressions (Stede et al., 1998, Endriss, 1998). All these different tasks are strongly interdependent. For example, in order to distinguish between certain dialogue acts it is necessary to compare date expressions. Dialogue act information is, in its turn, very important for the disambiguation process. This kind of knowledge based disambiguation is only integrated in the **deep** translation path. The German word "Essen", for example, can be translated into English as either "dinner" or "lunch", depending of the relevant time of day. Another German example is "vorziehen", which has two alternative readings, namely, "move" and "prefer". In order to use disambiguation as an additional information source for the selection procedure, we have assembled a set of ambiguities which are normally dealt with incorrectly by all translation paths except for **deep** (which is the only one that performs the above mentioned disambiguation procedures). When such ambiguities occur, the confidence value for **deep** is artificially increased.

## 5 Quality of Service

Selecting a certain translation path for a given segment has a significant impact on the translation quality, especially when the different translation paths significantly differ from one another. Translation quality is indeed one of the critical aspects as far as user acceptance is concerned. Additionally, there are other aspects of the quality service of automatic translation, which are important for user acceptance as well. In this section we introduce a set of dimensions of Quality of Service (QoS), and sketch the modification to our selection algorithm that enable us to support them. Analogously to QoS in Open Ditributed Programming (ODP), we can distinguish between the following main categories: timeliness, volume, and reliability.

**Intelligibility** This QoS dimension term was used in Pfeifer and Popescu-Zeletin (1996) in the context of media conversions, which are useful for unified messaging systems, for example. If one wishes to access the content of a fax over the telephone, a fax to speech conversion is necessary, which requires an optical character recognition module, and a speech synthesis module. For each conversion step, a QoS intelligibility is defined. In the case of the Verbmobil system, we use the more specific term *translation quality*.

**Delay/Incrementality** We define delay as the time from the beginning of speaking until the beginning of acoustic output from the system. This measure is influenced

by the size and buffering of the increments, which are processed by the system. Delay is a dimension of the timeliness category.

**Realtimefactor (RTF)** We define the RTF to be the quotient:
$RTF = (processing\_time)/(speaking\_time)$, where processing time is the time from the beginning of speaking to the end of acoustic output. Processing time covers the speaking time, which makes sense because the Verbmobil system is based on an incremental architecture, i.e. the processing starts in principle at the very moment that the input begins. RTF is a dimension of volume. We regard RTF as a specialization of the term *throughput*, which is more commonly used in multimedia applications.

**Loss Rate** Loss rate is the probability that given an input utterance, a translation is at all being generated. Loss rate is a dimension of reliability.

### 5.1 Level of Service

The term *Level of Service* is orthogonal to all the above described QoS dimensions. This term refers to the certainty that a required service quality can be obtained. In the case of language processing, the service quality cannot be deterministically guarantied. Thus, *Level of Service* refers to the probability that a required service quality would be obtained.

### 5.2 QoS Mapping

Two further points are important for the Quality of Service specification. Firstly, QoS parameters are naturally specified on the application level. The application is composed from a set of interacting modules, each module having in itself some module specific parameters. In order to provide QoS on application level, a mapping to the module specific parameters is necessary. Secondly, it should be noted that the parameters are not independent from one another. We motivate both points using the example of the disambiguation module. Consider the following utterance:

```
A) <Is it possible for you on the fifth of May ?>
B) <Ja> |  <Da kann ich leider nicht>
B') <Ja> | <Das geht>
```

The utterances B und B' are two possible replies to the question A. Each reply consists of two segments, the first of which - 'Ja' - requires disambiguation. In the first case, 'Ja' is an uptake particle, which should be omitted from the translation. In the second case 'Ja' can be simply translated to 'Yes'. For such disambiguation, it is necessary to compute the dialogue act of the *following* segment, which implies that the disambiguating module has to buffer a segment. If no buffering is permitted, only one default translation can be produced for both cases, resulting in poor translation quality. The quality of disambiguation is therefore dependent on incrementality. Disambiguation Quality is a module specific QoS parameter, which is one

component of the system level QoS parameter Translation Quality.

### 5.3 Dimensions of QoS in the Selection Module

**Translation Quality** The selection algorithm, along with the learning cycle that precedes it, are an attempt to maximize this parameter. Translation Quality is approximated by the Offline Evaluation Function.

**Delay/Incrementality** So far we have described an algorithm in which all translated segments within a given dialogue turn are expected to be present, before the selection itself can take place. This implies a relatively long delay, because the biggest possible increment unit, i.e. the whole turn, is being used. The maximal incrementality, and therefore the minimal delay, are achieved when the first ready segment is being chosen at each point. This implies, however, a deteriorated Translation Quality and an increased Loss Rate. The latter is due to the fact that the translation modules produce independent segmentations of the input utterance, that are likely to differ from one another. It is therefore not always possible to continue a certain segment with segments from other modules. Selecting a segment as soon as it is delivered by the translation module increases the risk that no continuation would be found, and hence the increased Loss Rate. Incrementality can be parameterized, if one decides to select a segment as soon as $n$ translation modules have delivered segments with similar segmentations ($1 \leq n \leq 4$).

**Loss Rate** The relation between Loss Rate and Delay has already been described in the previous section. The attempt to maximize the Translation Quality does not imply that when all alternative translations all under a certain quality threshold no output would be generated, but rather, that the user would be prompted to repeat their input. For the time being, this is only implemented for the case that all modules simultaneously fail to deliver any translation.

**RTF** In order to support conformance to the RTF specification for the translation service, the selection module supports a special QoS signal interface. A QoS management module monitors the runtime behavior of the translation modules, and sends a signal to the selection process if the estimated RTF is expected to exceed the specification. Note that the complete RTF can only be estimated in the course the selection process, because the speech synthesis runtime is not known in advance. After receiving a signal from the QoS management module, the selection process generates an output as soon as sufficient translation segments have been delivered.

### 5.4 Selection and Level of Service

The probability that the translation of the highest quality would be selected is highly dependent on the quality of the confidence values themselves. If confidence values do not correlate well enough with the translation quality of their corresponding segments, the selection algorithm is directly influenced, and the probability that high quality segments would be selected decreases accordingly.

## 6 Conclusion

We have described some of the difficulties that arise from the attempt to integrate multiple alternative translation paths, and to choose their optimal combination into one 'best' translation. Using confidence values that originate from different translation modules as our basic selection criterion, we have introduced a learning method which enables us to perform the selection in close to maximal accord with decisions taken by human annotators. We have described the problematic aspects of translation evaluation as such, and some of the strategies that we adopted for overcoming these difficulties. We have mentioned some additional sources of information that are used within our selection module, and also described the way in which it supports quality of service parameters. The extent to which this module succeeds in creating higher quality compound translations is of course highly dependent on the appropriate assignment of confidence values, which is performed by the various translation modules themselves. As a rule of thumb for evaluating our module's success we have formulated the requirement that its performance should be evaluated as better than the best single translation module. Recent Verbmobil evaluation results are based on annotating five alternative translations for a chosen set of dialogue-turns. The translations provided by the four single translation paths, and the combined translation delivered by the selection module, were all marked by the annotators as 'good', 'intermediate', or 'bad'. Judged by the percentage of 'good' turns from the overall number of annotated turns, the selection module shows an improvement of 27.8% compared to the best result achieved by a single module.

## References


Charniak, E., and Goldman, R. (1991) A probabilistic model of plan recognition. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 160–165.

Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Kipp,M., Koch, S., Maier, E., Reithinger, N., Schmitz, B., Siegel, M. (1997) Dialogue Acts in VERBMOBIL-2 *Second Edition* DFKI Saarbrücken, Universität Stuttgart, Technische Universität Berlin, Universität des Saarlandes, Verbmobil-Report 226.

Agmon, S. (1954) The relaxation method for linear inequalities. Canadian Journal of Mathematics, 6:382-392.

Amaldi, E., Mattavelli, M. A combinatorical optimization approach to extract piecewise linear structure from nonlinear data and an application to optical flow segmentation. TR 97-12, Cornell Computational Optimization Project, Cornell University, Ithaca NY, USA.

Buckow, J., Batliner, A., Gallwitz, F., Huber, R., Nöth, E., Warnke, V., and Niemann, H.. Dovetailing of Acoustics and Prosody in Spontaneous Speech Recognition In *Proceedings of the International Conference on Spoken Language Processing* volume 3, 571-574, Sydney, Australia.

Buschbeck-Wolf, B. Resolution on Demand. Universität Stuttgart. Verbmobil Report 196.

Cormen, T., Leiserson, C., Rivet, L. Introduction to Algorithms. MIT Press, Cambridge, Massachusetts.

Dorna, M. The ADT Package for the Verbmobil Interface Term Universität Stuttgart. Verbmobil Report 104X.

Emele, M., Dorna, M. Efficient Implementation of a Semantic-based Transfer Approach In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96).*

Emele, M., Dorna, M. Ambiguity Preserving Machine Translation using Packed Representations. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-ACL '98)*, Montreal, Canada.

Endriss, U. Semantik zeitlicher Ausdrcke in Terminvereinbarungsdialogen Verbmoil Report 227, TU Berlin.

Frederking, R., Nirenburg, S. Three Heads are Better than One. ANLP94P, 95-100.

Kilger A., Finkler, W. Incremental Generation for Real-Time Applications. DFKI Report RR-95-11, Research Center for Artificial Intelligence - DFKI GmbH.

Kipp M., Alexandersson, J., Reithinger, N. Understanding Spontaneous Negotiation Dialogue. In *Proceedings of the IJCAI Workshop Knowledge and Reasoning in Practical Dialogue Systems* Stockholm, Sweden.

Koch, S., Küssner, U., Stede, M., Tidhar, D. Contextual reasoning in speech-to-speech translation In *Proceedings of 2nd International Conference on Natural Language Processing (NLP2000).* Springer LNAI.

Küssner, U. Applying DL in Automatic Dialogue Interpreting. In *Proceedings of the International Workshop on Description Logics (DL-97)* 54-58. Gif sur Yvette, France.

Küssner, U. Description Logic Unplugged. In *Proceedings of the International Workshop on Description Logics (DL-98)* 142-146. Trento, Italy.

Motzkin, T.S., Schoenberg, I.J. The relaxation method for linear inequalities Canadian Journal of Mathematics, 6:393-404.

Och, F., J., Tillmann, C., Ney, H. Improved Alignment models for Statistical Machine Translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.* University of Maryland.

Pfeifer, T., Popescu-Zeletin, R. Generic Conversion of Communicating Media for supporting Personal Mobility In *Proceedings of the Third Cost 237 Workshop: Multimedia Telecommunications and Apllications*

Ruland, T., Rupp, C.J., Spilker, J., Weber, H., Worm, C. Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language. In *Proceedings of ICSLP 1998*.

Schmitz, B. Pragmatikbasiertes Maschinelles Dolmetschen. Dissertation, FB Informatik, TU Berlin, 1997.

Schmitz, B., Quantz, J.J. Dialogue Acts in Automatic Dialogue Interpreting. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*

Stede, M., Haas, S., Küssner, U. Understanding and tracking temporal descriptions in dialogue. In *Proceedings of KONVENS-98*.

Wahlster, W. Verbmobil: Translation of face-to-face dialogues. In *Proceedings of the Third European Conference of Speech Communication and Technology, Berlin*.

Worm C., Rupp, C.J. Towards Robust Understanding of Speech by Combination of Partial Analyses In *Proceedings of ECAI 1998*