The Janus-III Translation System

Speech-to-Speech Translation in Multiple Domains

Lori Levin, Alon Lavie, Monika Woszczyna, Donna Gates, Marsal Gavaldà, Detlef Koll and Alex Waibel
Carnegie Mellon University

Abstract.
The JANUS-III translation system translates spoken languages in limited domains. The current research focus is on is expanding beyond tasks involving a single semantic domain. The system combines translation components from multiple semantic domains into a unified system using multi-domain parse lattices. This approach yields solutions to several problems including ambiguity resolution, segmentation of spoken utterances into sentence units, modularity of system design, and re-use of earlier systems with incompatible output.

Keywords: speech translation, robust parsing, semantic grammars, multi domain, SOUP, JRTk

1. Introduction

Spoken Language Translation (SLT) systems have broken many barriers in the 1990's. Translation of well-formed, read speech with a small vocabulary has been replaced with translation of possibly ill-formed, spontaneous speech with a large vocabulary. A remaining limitation for SLT is that it is usually confined to a particular semantic domain. In this paper we address a step in the direction of domain independence — not completely free conversation, but integration of multiple limited domains, which at least gives speakers the option of discussing several related topics. Our system combines translation components from multiple semantic domains using multi-domain parse lattices. This approach yields solutions to several problems including managing a large multi-domain search space, segmentation of spoken utterances into sentence units, modularizing system design, and re-using components with incompatible output.

The JANUS-III speech translation system focuses on the broad domain of travel planning, scaling up from the JANUS-II domain of appointment scheduling (Spontaneous Scheduling Task or SST). Travel planning is still limited, but is significantly more complex than SST. The scheduling scenario naturally limits the vocabulary to about 3000 words in English and about 4000 words in Spanish and German, which have more inflection. The English vocabulary of our travel planning system is 10,000 words. The types of dialogues in SST are also naturally limited. A scheduling dialogue typically consists of opening greetings, followed by several rounds of negotiation on a time, followed by closings. Travel planning has more types of interactions. In addition to negotiations, openings, and closings, the travel domain includes information seeking, instruction giving, and dialogues that accompany non-linguistic domain actions such as paying and reserving. Finally, the main difference between SST and travel planning that we focus on in this paper is that travel planning contains a number of semantic sub-domains — for example, hotel accommodation, events, transportation — each of which has a number of sub-topics such as time, location, and price.

In scaling up from a single domain to a multi-domain system, we have concentrated on four problems. First, we had to coordinate the work of multiple grammar writers each working on different sub-domains. The grammar writers need to avoid duplication of effort on common phrases such as time expressions and must also maintain complete consistency with each other. A second problem concerning grammar development is how to re-use grammars that were written for other systems with different output requirements. Specifically, we had grammars from SST and from a car navigation task that were relevant to the travel planning domain. These grammars were written before the standardization of the interlingua representation for the travel planning domain and were producing incompatible output. Nevertheless, re-writing them would be a major effort. The third problem we encountered in our multi-domain system was managing the parser's search space. We are using a robust parser for spoken language that can parse fragments of utterances, possibly overlapping. Adding to this the extra interpretations of fragments in multiple domains (e.g., interpreting *six thirty* as a room number and flight number as well as a time) results in a search space of a significantly larger scale than it would be for a single domain. The modular system design with multi-domain parse lattices that we describe in this paper addresses these three issues. A fourth issue, general resolution of ambiguity using discourse context was a topic of our previous research on SST (Levin et al. 1995, Qu et al. 1996a, Qu et al. 1996b, Rosé et al. 1995, Lavie et al. 1996) and is not covered here.

The remainder of this paper is organized in the following way: We begin with an overview of our translation system in Section 2. Section 3 is about the JANUS Recognition Tool Kit (JRTK) for speech recognition, and is self-contained so that it can be skipped by readers who are not interested in the details of speech recognition. Section 4 is devoted to the SOUP parser, our main analysis component. Section 5 describes the newly developed interlingua representation we use for the C-STAR

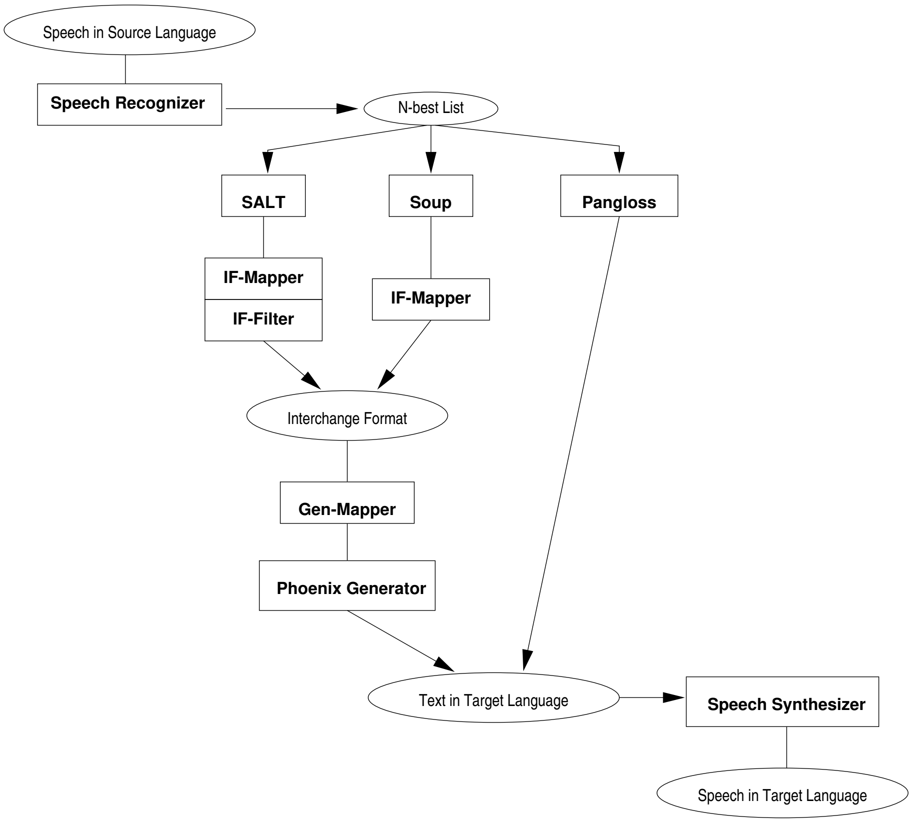

Figure 1. Components of the Translation System

Travel Planning domain. Section 6 focuses on the engineering aspects of expanding our system to multiple domains. End-to-end system evaluation and some recent performance results are described in Section 7. Finally, our main current and future research topics are discussed in section 8.

2. System Overview

A component diagram of the current JANUS speech translation system for the travel domain can be seen in Figure 1. The main system modules are speech recognition, machine translation, and speech synthesis. The interface between the speech recognizer and the translation system is via an N-best list of text string hypotheses in the source language. At the end of the translation process, a speech synthesizer converts the target language text into speech. We currently use FESTIVAL (Black and

```
Input:  WE HAVE TWO HOTELS AVAILABLE

Parse Tree:  [give-information+availability+hotel]
                ( we have [hotel-type=]
                            ( [quantity=] ( two )
                              [hotel] ( hotels ) )
                          available )

IF:          a:give-information+availability+hotel
                (hotel-type=(hotel, quantity=2))

Gen. Input:  [a==give-information+availability+hotel]
                ( [hotel-type=] ( [hotel] ()
                                  [quantity=] (2)))
```

Figure 2. Parsing Example

Taylor 1997, Black et al. 1998), a speech synthesis system developed at the University of Edinburgh.

Our current system includes three separate translation chains. Our main translation module is an interlingua-based approach that uses rule-based components for both analysis and generation. However, we have recently been experimenting with two alternative translation modules. The first is an interlingua-based module in which analysis is partially performed by a statistical parser (instead of the rule-based parser). The second approach is a direct translation approach that primarily uses example-based machine translation (EBMT). This translation module was originally developed for the PANGLOSS and DIPLOMAT projects (Frederking et al. 1997, Frederking et al. 1998, Nirenburg (ed.) 1995), and has been adapted for the C-STAR domain. The two alternative translation chains are further discussed in Section 8.

In the interlingua-based translation modules, translation is performed by analyzing the source input string into an interlingua representation, and then generating a string in the target language. In our main analysis sub-module, the input string is analyzed by SOUP, a robust parser designed for spoken language. SOUP, described in detail in section 4, works with semantic grammars in which the non-terminal nodes represent concepts and not syntactic categories. The output of the parser represents the meaning of the input and serves as an interlingua for translation. The Parser-to-IF mapper then converts this representation into a canonical *Interchange Format* or IF (see Section 5) that was jointly designed by the C-STAR consortium member groups. The mapper performs a simple format conversion, and does not contribute any significant information beyond that derived by the parser.

The IF interlingua representation is then passed on to generation, which generates output text for several different target languages (currently English, German and Japanese) using target language generation grammars. Note that this framework supports generation back into the source language (in our case, English), which results in a paraphrase of the input. This provides the user with a mechanism for verifying analysis correctness, even when he/she is not fluent in the target language. The IF can also be exported to the generation systems of other C-STAR partners for translation into languages not supported at CMU (French, Italian, and Korean). The generation process first uses a generation mapper, which converts the IF into a tree semantic representation which is then passed on to the generation module. The PHOENIX generator then produces a string in the target language.

The analyzer and generator are language-independent in that they consist of a general processor that can be loaded with language specific knowledge sources. Our travel domain system currently includes analysis grammars for English and German and generation grammars for English, German, and Japanese. Additional languages (Spanish and Korean) are available for sentences in the scheduling domain. Figure 2 shows an example of the output from the parser, the IF produced by the analysis mapper and the output from the generation mapper.

2.1. Semantic Grammars

An important feature of JANUS MT is the use of semantic grammars. Semantic grammars describe the wording of concepts instead of the syntactic constituency of phrases. For example, a semantic grammar indicates that the wordings *we have* or *there are* express availability for rooms, flights, and other concepts. There were several reasons for choosing semantic grammars. First, task-oriented domains such as travel planning lend themselves well to semantic grammars because there are many fixed expressions and common expressions that are almost formulaic. Breaking these down syntactically would be an unnecessary complication. Additionally, spontaneous spoken language is often syntactically ill formed, yet semantically coherent. Semantic grammars allow our robust parsers to scan for the key concepts being conveyed, even when the input is not completely grammatical in a syntactic sense. Furthermore, we wanted to achieve reasonable coverage of the domain in as short a time as possible. Our experience has been that, for limited domains, 60% to 80% coverage can be achieved in a few months with semantic grammars.

Although we have been happy with our choice of semantic grammars, there are some draw-backs. Semantic grammars are not easily adapted to new domains, whereas Syntactic grammars can be re-used easily in new domains because the syntactic categories remain constant. Furthermore, although semantic grammars are ideal for task-oriented sentences such as making reservations, giving prices, etc., they are not well suited for descriptive sentences in the travel domain such as *The castle was built in the thirteenth century* or *The temple has a beautiful garden*. In Section 6 we describe how our method for multi-domain integration addresses the portability of semantic grammars by providing shared sub-grammars and allowing for parses from a domain specific grammar to be combined with parses from more general grammars with cross-domain applicability (e.g., for times and dates) in a shared parse lattice. This method makes it possible to keep using semantic grammars for task-oriented sentences in future versions of our system. However, we do expect in the future to use syntactic grammars for descriptive sentences.

3. Speech Recognition

3.1. JRTk

For the speech recognition in the JANUS speech-to-speech translation system, we use the JANUS Recognition Toolkit, JRTk. As implied by the name, JRTk is a toolkit that can be programmed to build a variety of dedicated recognition systems. The programming interface is realized as an integrated tcl interpreter, used to run scripts from which the application developer can create and use the objects that make up the recognizer. The flexibility of this toolkit makes it relatively easy to build a recognizer that is tuned to optimal performance for a multi domain speech translation task.

3.2. Speech Recognition Components

The goal of speech recognition for speech-to-speech translation is to produce one or several hypotheses that are as close as possible to what the speaker said. This output depends on the recorded speech signal and additional world knowledge. We have to find the word sequence W for which the probability

$$P(W|A) = \frac{p(A|W)P(W)}{p(A)} \tag{1}$$

is largest.

Here, $p(A)$ is the probability of observing the recorded signal. Because it is independent of the word sequence W, it can be ignored in the maximization. $P(W)$ is the *a priori* probability of the word sequence, and is independent of the actual input signal. It is in $P(W)$ that we try to capture most of the world knowledge. The model used to estimate $P(W)$ is usually referred to as the *Language Model*. Finally, $p(A|W)$ is the probability of observing the signal A under the assumption that the actual word sequence is W. The model used to estimate $P(A|W)$ is called *Acoustic Model*.

It is important to understand that a considerable number of simplifications and approximations are required to make this maximization problem traceable on today's computers. Therefore, the word sequence with the highest approximated probability will usually not be the same as a human transcription of the original utterance. For common benchmark tasks, the number of errors in a sequence of 100 words of input speech ranges between five (for simple tasks) and 50 (for fast, spontaneous telephone speech with strong coarticulation).

3.3. Acoustic Models

For recognition purposes, the speech signal A is usually represented as a sequence of feature vectors extracted from the original speech input. The goal of the *Acoustic Model* is now to compute the probability $p(A|W)$ for observing these vectors under the assumption that the actual utterance consisted of the word sequence W. The words are cut into smaller segments, assigning the same symbol to units that 'sound alike'. A common set of such units are the *phonemes*. Since for many languages it is difficult to derive the sequence of phonemes from the spelling of a word, a pronunciation dictionary is an important knowledge source for a speech recognition system.

```
quit        K W IH T
quite       K W AY T
to          T UW
too         T UW
two         T UW
```

Figure 3. Examples taken from a JRTk pronunciation dictionary

When expanding a recognizer to a different domain, new words have to be added to the dictionary. For the JRTk recognizer used in our Janus system, these dictionaries are compiled by a mixture of manual input and automatic generation and verification of pronunciation variants based on recorded examples of the word in a number of different utterances. In the travel domain, the often inconsistent pronunciation of foreign names and places (e.g. Schloßstraße, Gion) presents a special problem that is subject to ongoing research.

3.3.1. *Domain Independence*

Since the sound of a phoneme changes depending on the adjacent phonemes, the phonemes are commonly subdivided into three segments that are modeled depending on the identity (e.g, UW or T) or kind (vowel or stop) of the surrounding phonemes.

The more detailed models a system has, the better it will work for the task it was trained on. Acoustic models with a total of more than 360,000 Gaussians are often used in speech recognition evaluations. However, such detailed models are slow to compute and do not generalize enough if many new words have to be added to the dictionary when expanding to new domains.

To provide domain independence, the acoustic models used for our multi-domain ST system have a total of 64,000 Gaussians. Furthermore, they have been trained on a combination of data collected for the travel domain and data from other tasks such as read newspaper data. A number of techniques like the generalized Bucket Box Intersection Algorithm (Woszczyna, 1998) were developed to allow real-time recognition with acoustic models of this size.

3.4. Language Models

The language model is used to compute the a priori likelihood for a sentence based on statistical knowledge derived from transcribed dialogues and related texts.

The JRTk based recognizer in our Janus speech-to-speech translation system uses *trigrams* to estimate the probability of $P(W)$.

When porting to new domains, providing enough data for language modeling is one of the most important problems. If only a limited amount of data is available, that data can be used to find similar sections in more abundant text sources, such as newspaper text. These sections are weighted with their similarity to the example data and then used to build a full language model.

4. The Soup Parser

The main analysis component in our system is the Soup parser, which was specifically designed for real-time analysis of spoken language utterances with very large, multi-domain semantic grammars. The Soup

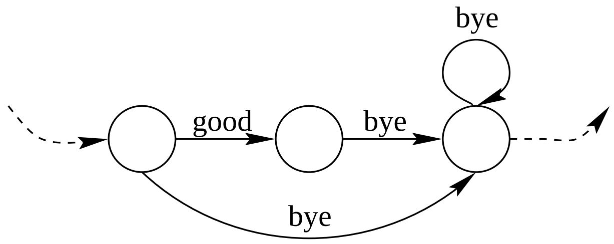

Figure 4. PRTN for the RHS sequence `( *good +bye )`

parser was inspired by Wayne Ward's PHOENIX parser (Ward, 1990) and is a stochastic, chart-based, top-down parser of context-free grammars (CFGs).

4.1. Grammar Representation

Internal to the parser, a CFG is represented as probabilistic recursive transition networks (PRTNs). For example, the PRTN in Figure 4 represents the right hand side sequence `( *good +bye )`, with each arc annotated with a probability, so that the probabilities of each node's outgoing arcs sum to one. Grammar arc probabilities are initialized to the uniform distribution but can be perturbed by a training corpus of correct parses: given a set of desired (but achievable with the given grammar) parse trees, the training procedure increments counts and adjusts probabilities on the PRTN nodes and arcs along the path that leads to the desired parse. Given the direct correspondence between parse trees and arc paths along the grammar, training can be conducted in a very efficient manner. Arc probabilities are included in the heuristic function that is used to guide the search in the parsing stage. More likely paths are thus preferred and explored first.

4.2. The Parsing Algorithm

Given a grammar and an utterance to be analyzed, SOUP's task is to provide a ranked list of *interpretations* of the utterance according to the grammar, where each interpretation is a sequence of non-overlapping *parse trees*, and a parse tree can be seen as a traversal of the CFG, i.e., a path through the PRTNs, starting at a top-level non-terminal and covering a portion of the utterance. Words *between* identified parse-trees may be skipped or ignored. The parsing process is accomplished by (a) populating a chart of completed constituents, and (b) finding the "best" sequence of combinations.

The chart is a three-dimensional matrix that is dynamically allocated on a strict on-need basis. The three axes correspond to (i)

non-terminal ID, (*ii*) start position, and (*iii*) end position. The search proceeds in a top-down fashion, attempting to match all top-level non-terminals at all positions of the input. Internally, SOUP constructs *parse DAGs* (directed acyclic graphs) rather than parse trees. This allows efficient representation of ambiguities, similar to the idea of *shared parse forests* (Tomita, 1986).

At the end of the parsing process, the chart contains a lattice of complete (possibly overlapping) parse trees, each covering a portion of the input. The ranked list of utterance interpretations is then created from the parse lattice. This is done using the following set of disambiguation heuristics:

1. **Maximize coverage:** Given two interpretations, prefer the one that covers the highest number of input tokens.
2. **Minimize number of parse trees**. Given two interpretations, prefer the one that has fewer parse trees. The rational behind this principle is to try and minimize parse fragmentation.
3. **Minimize the number of parse tree nodes:** Given two interpretations, prefer the one that has fewer parse tree nodes.
4. **Minimize the number of wildcard matches:** Given two interpretations, prefer the one that has fewer usages of the wildcard symbol (`_$any$_`).
5. **Maximize the probability of parse trees as paths along grammar arcs:** Given two interpretations, prefer the one with higher path probability. The path probability is computed to be the average of the arc probabilities along the arcs employed in the construction of the parse tree.
6. **Maximize probability of subdomains of the parse tree sequence:** Given two interpretations, prefer the one with a higher probability of subdomains. See section 4.3 below for a detailed description of this heuristic.

We have been experimenting with several linear combinations of the above set of heuristics, including combinations which apply the heuristics in ranked order (as above). Although experiments are still in progress, the strict ranking is already highly effective for correct disambiguation.

4.3. Disambiguation with Statistical Domain Knowledge

Since Soup uses multiple domain grammars, each parse tree in an interpretation may be "drawn" from a different subdomain. We would thus like to use probabilistic information about the likelihood of the subdomains given the input to assist in the disambiguation process. Since each parse tree has a unique subdomain to which it belongs, given a set of alternative interpretations, our goal is to find the interpretation that has the most likely sequence T of subdomains given the sequence of input words $\mathbf{W}$, i.e. to maximize the probability $P(\mathbf{T}|\mathbf{W})$, where $T = (t_1, t_2, \cdots, t_k)$ is the sequence of subdomains corresponding to the sequence of parse trees in the interpretation.

$$P(\mathbf{T}|\mathbf{W}) = \frac{P(\mathbf{W}|\mathbf{T}) \cdot P(\mathbf{T})}{P(\mathbf{W})} \tag{2}$$

Since $P(\mathbf{W})$ does not change once the utterance has been recognized, this is the same as maximizing $P(\mathbf{W}|\mathbf{T}) \cdot P(\mathbf{T})$. To simplify the computation, we assume that the probability of the words W_i covered by parse tree i, depend only on the domain t_i to which the parse tree belongs. Thus, $P(\mathbf{W}|\mathbf{T}) = \prod_i P(W_i|t_i)$. To estimate $P(W_i|t_i)$ we use a unigram model, where the frequency of observing each word in the vocabulary for each subdomain is calculated from a tagged training database. The probability for the sequence of domains $P(\mathbf{T})$ within one utterance is approximated by a unigram or a bigram statistic:

$$\begin{aligned} P(\mathbf{T}) &\approx P(t_1) \cdot P(t_2) \cdot \ldots \cdot P(t_N) \\ &\approx P(t_1) \cdot P(t_2|t_1) \cdot \ldots \cdot P(t_N|t_{N-1}) \end{aligned} \tag{3}$$

4.4. Real-World Considerations

There has been a continued effort to tailor Soup to the practical needs of real-world grammars (i.e., very large) and real-world grammar development (i.e., a team effort). For example, the current combined grammar for English scheduling and travel planning, contains on the order of 5K nonterminals, 18K rules, and 8K lexical entries, giving rise to a collection of PRTNs in the order of 39K nodes and 73K arcs.

For more efficient grammar development, we have constructed a graphical grammar editor, called G-Soup, that allows for: (1) Graphical visualization, creation, deletion and editing of nonterminals and rules; (2) Automatic assessment of rule coverage; (3) Automatic detection of rule conflicts; and (4) Automatic and manual annotation of rules.

Performance-wise, Soup, implemented in C++, is very efficient. On an English grammar for the scheduling task, containing 600 concepts (21 top-level, 466 auxiliary), 2880 grammar rules and 829 lexical entries, which give rise to 6373 grammar nodes and 10480 grammar arcs, running on a Sun-Ultra-I at 167 MHz, a set of 609 sentences containing a total of 5502 words were parsed in 4352 ms, i.e., at an average of 7.146 ms per sentence, or almost 140 sentences per second.

Soup has also been extended in some novel ways to handle semantic grammars for multiple domains. These are described in detail in Section 6.

5. The C-STAR Interchange-Format

The Janus project has chosen an interlingual approach to multi-lingual translation in the context of the C-STAR consortium. Interlingual machine translation is convenient when more than two languages are involved because it does not require each language to be connected by a set of transfer rules to each other language in each direction (Nirenburg et al. 1992). Adding a new language requires only writing one analyzer, mapping utterances into the interlingua, and one generator, mapping interlingua representations into sentences, and results in all-ways translation between all languages. A consequence of this is for the C-STAR consortium is that each partner implements analyzers and generators for its home language only. There is no need for bilingual teams to write transfer rules connecting two languages. A further advantage of the interlingual approach is that it supports a paraphrase option for monolingual MT users. Users' utterances are analyzed into the interlingua and then generated again in the user's language from the interlingua. This allows the users to confirm that the system produced correct interlinguas for their utterances.

The main principle guiding the design of the interlingua is that it must abstract away from peculiarities of the source languages in order to account for MT divergences and other non-literal translations (Dorr, 1992; Levin and Nirenburg, 1994). In the travel domain non-literal translations may be required because of many fixed expressions that are used for activities such as requesting information, making payments, etc.

An additional factor that constrains interlingua design in the C-STAR consortium is that it is used at multiple research sites. It was therefore necessary to design a simple interlingua that could be used reliably by many MT developers. Simplicity is possible largely because we are working on travel planning, a task-oriented domain. In a task-oriented domain, most utterances perform a limited number of *domain actions* (DAs) such as requesting information about the availability of a hotel or giving information about the price of a hotel. These domain actions form the basis of the C-STAR interlingua, which is known as the *interchange format*, or IF.

A DA consists of three representational levels: the *speech act*, the *concepts*, and the *arguments*. In addition, each DA is preceded by a speaker tag (`a:` for agent or `c:` for customer) to indicate who is speaking. The speaker tag is sometimes the only difference between the IFs of two different sentences. For example, *Do you take credit cards?* (uttered by the customer) and *Will you be paying with a credit card* (uttered by the agent) are both requests for information about credit cards as a form of payment. Plus signs separate speech acts from concepts and concepts from each other. In general each DA has a speaker tag and at least one speech act optionally followed by a string of concepts and/or a string of arguments. DAs can be roughly characterized as shown in (4). However, there are constraints on the order of concepts so that not all combinations are possible. There are approximately 26 speech acts, 100 concepts, and 150 arguments. (Exact numbers will be in the final version of the paper.)

(4) $speaker : speech\ act\ {+concept}^{*}\ argument^{*}$

In example (5) the speech act is `give-information`, the concepts are `availability` and `room`, and the arguments are `time` and `room-type`. The possible arguments of a DA are determined by inheritance through a hierarchy of speech acts and concepts. In this case `time` is an argument of `availability` and `room-type` is an argument of room. Example (6) shows a DA which consists of a speech act with no concepts attached to it. The argument `time` is inherited from the speech act `closing`. Finally, example (7) demonstrates a case of DA which contains neither concepts nor arguments.

(5) On the twelfth we have a single and a double available.
```
a:give-information+availability+room
(room-type=(single & double),time=(md12))
```

(6) And we'll see you on February twelfth.
```
a:closing (time=(february, md12))
```

(7) Thank you very much
```
c:thank
```

These DAs do not capture all of the information present in their corresponding utterances. For instance they do not represent definiteness, grammatical relations, plurality, modality, or the presence of embedded clauses. These features are generally part of the formulaic, conventional ways of expressing the DAs in English. Their syntactic form is not relevant for translation; it only indirectly contributes to the identification of the DA.

6. Engineering a Multi-domain System

As already mentioned earlier, semantic grammars are very attractive for the analysis of spoken language input. For limited domains, semantic grammars are fairly fast to develop and fairly easy to maintain. However, they are usually hard to expand to cover new domains. New rules are required for each new semantic concept, since syntactic generalities cannot usually be fully utilized. For large domains, this can result in very cumbersome grammars that become difficult to expand and further develop, and which are highly ambiguous in nature. In our current system, significant effort has been put into addressing these difficulties via modularization of the grammars and enhancements to the parsing architecture that allow it to support the integration of multiple domain grammars and interlingua representations.

6.1. Grammar Modularization

Modularization and common libraries have long been a well-established concept in software development. Many of the advantages of these concepts similarly apply to the task of engineering large semantic grammars. Whereas in software engineering the goal is to divide up the overall program into well defined modules that can be separately developed and maintained, we wish to similarly divide the task of grammar development into well defined sub-grammars that can be developed and maintained independently, while sharing common sub-grammar portions via a grammar library. This requires some engineering in the design of the overall grammar. To reap the benefits of modularization, the grammar must be defined in a compositional fashion. In many cases, a large semantic domain can be divided into smaller sub-domains in a fairly straightforward way. Each of the sub-domain grammars build upon lower-level concepts, some of which are likely to appear in more than one sub-domain (i.e. time and date expressions, expressions of request and desire, availability or non-availability, etc.). The analysis of these common concepts can thus be expressed via grammar rules that are drawn from a common library, which is then shared between the sub-domain grammars.

In our system, we divided the large Travel Planning domain into four main sub-domains: Hotel Information and Reservation, Transportation, Sights and Events, and General Travel (which captures general concepts related to the travel domain which do not fall naturally under the other sub-domains). Additionally, we defined a "cross-domain" grammar, which covers actions that are not specific to the travel domain, and are expected to occur in almost any spoken language task: greetings, formalities, expressions of understanding or misunderstanding, etc. Maintaining the cross-domain grammar as a separate grammar module should prove useful for reuse in other domains. We also constructed a shared grammar module to cover the lower-level concepts that are used in the various travel sub-domain grammars. These include time and date expressions (such as *around 5pm on Friday*) as well as lists of proper names (i.e. *Monika*). The main benefit from this modularization is in the substantial reduction in complexity of developing and maintaining the overall complete semantic grammar. Furthermore, The shared library and the cross-domain sub-grammar substantially reduce the effort required to expand the system to new domains.

6.2. Analysis with Multiple Domain Grammars

In parallel to the design of the modular collection of sub-domain grammars and shared grammar files, the SOUP parser was extended in order to allow it to support the integration of multiple domain grammars and interlingua representations in an elegant and efficient way. As in the case of a single domain system, the task of the analyzer is to analyze a spoken input utterance as a sequence of top-level concepts. In the multi-domain system, however, the sequence of top-level concepts may be from multiple domains. Thus, the union of top-level concepts from all domain and sub-domain grammars must be considered during parse-time. Working with multiple domain grammars also has a significant impact on the level of ambiguity, since there may be multiple ways to segment an input utterance into a collection of multiple domain top-level concepts.

Rather than running multiple parsers for the various domains, and then combining the output from the separate analysis units, we chose to modify the SOUP parser to effectively parse with multiple domain grammars concurrently. In effect, the parser works with a large "union" grammar that consists of the various separate domain grammars, tied together at the top-level concept level. Since the various domain grammars are developed independently, care must be given not to confuse concepts from different domain grammars that accidentally share the same name. Only concepts that are explicitly designed to be shared between the various grammars should in fact be common in the union grammar. SOUP handles this problem by attaching a tag to the concepts of each domain grammar when loading the set of separate domain grammars. For example, concepts originating from the Hotel Reservation domain grammar will all be tagged with a suffix `:HTL`. The actual tags used can be specified as parameters to the parser. Shared grammar files are uniquely identified to the parser at load time. Concepts in shared grammar files are tagged with a special tag, which is also used to tag any occurrences of the shared concepts in the various domain grammars. This allows all shared concepts to be accessible to all domain grammars.

Since each utterance is parsed as a sequence of top-level concepts, the parser also implicitly provides the segmentation of the utterance into concepts. Thus we do not need a separate program for segmenting spoken utterances into sentences. However, as mentioned earlier, this introduces a significant additional source of ambiguity, since utterances may often be segmented into sequences of top-level concepts in multiple ways. The efficient lattice representation used by the SOUP parser is effective in handling such high levels of ambiguity. The parse scoring heuristics (see Section 4) are used to produce a ranked N-best list of parses from the set of parses represented in the lattice. The N-best list is then re-scored using statistical domain information, as described in Section 4.3. Note that the sequence of concepts that comprise one utterance do not have to all originate from the same sub-grammar. The utterance in example (8) contains concepts (and sub-parses) from three different sub-domain grammars.

(8) Hello,
I would like to make a reservation for a flight to Frankfurt on the fifth
and maybe also book a hotel room.

```
(GTR) c:greeting
(TPT) c:request-action+reservation+temporal+flight
(HTL) c:request-action+reservation+features+room
```

Also note that the greeting "*hello*" is parsed by a cross-domain grammar, while the words "*Frankfurt*" and "*on the fifth*" are parsed by the shared grammar, in this case accessed by the sub-domain grammar for transportation.

A considerable advantage of our approach is that grammars producing different interlingua representations can be integrated into one system on the sub-utterance level. This is made possible by the fact that the parser works with a unified grammar that consists of distinguishable non-overlapping domain grammars. Grammars that were developed for

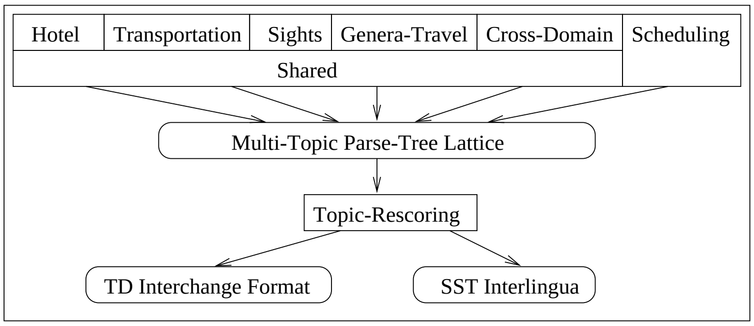

Figure 5. Combining multiple sub-domain grammars with shared and cross domain grammars

other domains can simply be appended, even if they produce a different output format. The tags that are associated with each of the domain grammars can then be used to identify the domain from which each top-level concept (and parse-tree) originated. Since each parse-tree is marked with a domain tag, it is easy to make sure that it is then handled by the appropriate mappers and generators. In our current system we combined the grammars developed for the travel domain with our previous grammars developed for the scheduling task. The interlingua representations for these two tasks are different, but the system can elegantly handle both. In the following example (9), the utterance is analyzed into two top-level concepts and interlingua representations, the first from the Hotel sub-domain (using an IF interlingua), while the second is from the scheduling domain (using its own interlingua representation).

(9) I would like to make a reservation for a hotel room – do you have time on Friday?
`(HTL) c:request-action+reservation+features+room`
`(SST) q_your_availability`

Figure 5 shows the grammar configuration we use to cover the large travel domain via several sub-domain grammars, and how the multiple domains are handled by our analysis module.

7. System Evaluation

To get realistic data to evaluate and improve our system, we conducted a series of user studies. The data from each study was first used to evaluate the system, then for error analysis and finally for development.

In addition to the results reported here, the subjects were also given a questionnaire on user interface issues, that was evaluated to improve HCI aspects of the system.

The subjects involved in all user-studies had little or no previous exposure to speech recognition or speech translation. They were seated in a moderately noisy office and asked to play the role of a traveller booking a trip to Germany or, in the case of the latest user study, to Japan. The travel agents (researchers from our group) were placed in a different office. The only means of communication between the "client" and the "agent" were:

– our speech-to-speech translation system, translating from English via IF to English

– our multi-modal interface allowing for handwriting recognition and sharing web-pages

– a muted netmeeting video-conference (no audio)

During the entire duration of the user study, the subjects were observed and videotaped by a researcher. Instructions on how to best use the system and interventions in case of problems were kept to a minimum.

Sentence based JANUS MT evaluations are run as end-to-end evaluations of translation output from speech input. Bilingual graders compare the source language input and target language output for each sentence. The grades assigned are OK, bad, and perfect. OK translations contain all the information from the source language sentence with no extra misleading information. Perfect translations meet this criterion and are, in addition, fluent in the target language. Our evaluation procedures are described in more detail in Gates et al. (1996). Table I reports the results of a recent evaluation. The evaluation was conducted on a set of 132 sentences, previously unseen by the grammar developers, each of which contains one or more DAs. The data was taken from our latest user study of subject trying to book a trip to Japan.

Experiment 1 in Table I shows the quality of the speech recognition output measured by the same criteria as the output of the translation engine — OK for retaining all relevant meaning and Perfect for being fluent. For about 22% of all utterances, some important change of meaning had occurred due to a recognition error in the best matching hypothesis. Preliminary experiments using word graphs rather than first best hypotheses indicate that for about half of these utterances even a small word graph contains a hypothesis of the correct meaning.

Experiments 2 and 3 give the performance of the system for paraphrasing back into English from transcribed text (Experiment 2) and

Table I. Translation Grades for English to English, English to Japanese, and English to German translation using the Soup parser.

	Method	Output Language	OK+Perfect	Perfect
1	Recognition only	English	78 %	62 %
2	Soup on Transcription	English	74 %	54 %
3	Soup on Recognition	English	59 %	42 %
4	Soup on Transcription	Japanese	77 %	59 %
5	Soup on Recognition	Japanese	62 %	45 %
6	Soup on Transcription	German	70 %	39 %
7	Soup on Recognition	German	58 %	34 %

Table II. Translation Grades for the example based English to German translation.

	Method	Output Language	OK+Perfect	Perfect
8	Pangloss on Transcription	German	80 %	36 %
9	Pangloss on Recognition	German	67 %	31 %

speech recognition output (Experiment 3). An error analysis showed that 8% of all utterances did not get a correct translation because of speech recognition errors. Another 20% of all utterances did not get correct translations because of coverage of the interchange format or grammars.

Experiments 4 and 5 give the performance for English-to-Japanese translation from transcribed English input (Experiment 4) and recognized English input (Experiment 5). The slightly better results in comparison to English-to-English paraphrase reflects the subjective nature of the grading process more than the actual performance. Experiments 6 and 7 report the numbers for English-to-German translation using the GenKit generator for German. The development time for the German generation grammar prior to the evaluation was extremely short (less than 4 months), resulting in lower coverage.

Table II reports the results for English-to-German translation using Pangloss example-based and direct translation. While the results look

Table III. Translation Grades for different development stages, English to English translation.

	Method	OK+Perfect	Perfect
January 99	Soup on Transcription	69 %	46 %
January 99	Soup on Recognition	55 %	36 %
April 99	Soup on Transcription	70 %	49 %
April 99	Soup on Recognition	57 %	38 %
August 99	Soup on Transcription	74 %	54 %
August 99	Soup on Recognition	59 %	42 %

encouraging, it is difficult to get high quality German output. However, Pangloss offers an excellent fall-back strategy for uncovered or out of domain utterances.

Table III shows the progress of the grammar development over the last six months.

At a first glance, these numbers of the sentence based evaluation seem to indicate poor system performance. However, the task completion rate is much higher than the sentence accuracy. If on average 30% of all sentences are not translated in an acceptable way, the chance of all sentences in a twenty sentence dialogue being translated completely correctly is less than 1%, but that does not imply that a 20 sentence dialogue has less than 1% chance of succeeding. Most subjects in the user studies we conducted achieved their prime dialogue goals, namely to book their flight and a hotel room, as well as getting some informations on local sights and events. Most users were able to overcome problems generated by recognition errors or lack of grammar expression coverage by rephrasing their request. The 30% sentence-level error rate indicates that on average one utterance out of three requires a second attempt in order for the translation to come across. Some secondary dialogue goals, like getting directions to a sushi restaurant near the hotel or obtaining a map of the train station had not been covered by the development of any of the systems components (speech recognition, grammars, IF, agent databases), and were therefore impossible to achieve. We are still working on a full task-based evaluation [24] that will include the percentage of dialogue goals that were met as well as the effort in terms of number of attempts required to meet them.

8. Current and Future Work

The current architecture framework of the JANUS MT engine described in this paper has provided us with a solid design foundation for developing our translation system for the travel domain, which has proven to be a challenging task. Much of our current work involves incremental improvements in the coverage of our grammars and other knowledge sources and adding new languages in preparation for a thorough end-to-end evaluation. We are also working, however, on a number of advanced extensions to the translation system itself. These include the analysis of more advanced statistical disambiguation techniques, and the development of several alternative translation methods that we intend to combine with our grammar-based approach.

Multi-Engine Translation: Multi-engine translation was proposed by Frederking et al. (1994) and has since been implemented in the Diplomat (Frederking et al., 1998) and Verbmobil systems. A multi-engine system applies multiple translation programs simultaneously and makes a translation by composing the best parts from the various outputs. Typically, a multi-engine system might include knowledge-based, statistical, and direct dictionary based approaches. In our case the components will be the knowledge based system described in this paper, statistical dialogue act assignment, and glossary lookups. A major research issue in multi-engine translation is improving methods for combining the outputs of the various engines (Frederking et al., 1998).

Combined Statistical/Grammar-based Analysis: One weakness of the grammar-based analysis system is that it is not very robust to concept phrasings that deviate significantly from those expected in the grammars, or to the occurrence of unexpected "noise" within concepts. To address this problem we are developing an alternative parsing method that combines both statistical and grammar information. Statistical information is used in order to identify the DA, in cases where the grammar fails to do so with reasonable confidence. Using constraints from the interlingua specification, we then predict the set of possible arguments that can occur with the DA. A modified version of the grammars for parsing just argument fragments is then used in order to extract the appropriate arguments from the utterance. Preliminary experiments with this method are showing encouraging results.

International Cooperation: The *interchange format* used for our travel domain system was designed to allow the integration of several translation systems of different sites into a larger distributed translation system. To test the quality of the IF definition, IF output from the systems of other sites (such as IRST in Italy) was run through the English an Japanese generators at CMU and vice versa. Also, first experiments in Japanese to English and English to Japanese translation have been performed with one system running at ATR in Japan and the other at CMU in the US. The results of these preliminary experiments are highly encouraging.

Task Based Evaluation: Our current sentence level evaluations measure the accuracy of translation, but do not show how mistranslations interfere with task success. Task based evaluations measure success in completing a task, in this case making travel reservations. Task based evaluations have been frequently applied to human-machine dialogue (Walker et al., 1997; Danieli et al. 1995) but less frequently to human-human dialogue mediated by machine.[1] In addition, our task is more complex than others that have undergone task based evaluation. Our speakers plan many aspects of a trip in one dialogue and may change goals frequently. The challenges posed by designing a task based evaluation for MT include tracking and tagging the speaker's changing goals and normalizing for speaker style when counting repair sentences.

Integration with Multimedia techniques: One possible scenario for the use of travel domain speech translation is a video-conference between a travel agent in a foreign country and the interested client. In such a scenario, it is desirable to have a vast array of additional tools for communication available. The travel agent should be able transmit pictures and videos of locations to the client, point to maps and transfer documents such as price lists. It also makes sense for the travel agent to access the agencies databases through the same interface that is used for the communication with the client, especially if the speech translation involved in this communication already provides speech understanding components that can also be used for database access. We are experimenting with the combination of a number of multimedia techniques such as speech, handwriting, face-tracing and gesture recognition for human-computer interfaces, that could also be applied to this scenario.

Acknowledgements

The IF formalism is the result of a close cooperation of the six C-STAR-II partners http://www.c-star.org. Siemens played an important role in devising the initial format and structure. The original description of the IF was written by Mirella Lapata. User studies were conducted by

[1] We believe that Verbmobil may have conducted task based evaluations, but were unable to find the references.

Alexandra Slavkovic. Part of the work on statistical DA identification was done by T. Fukada from ATR during a research term at CMU. Matthew Broadhead contributed some work on topic identification. The English speech recognition engine and grammars were developed using scheduling and travel domain data collected under the supervision of Sondra Ahlén. We would also like to thank our grammar writers Daniela Müller, Kavita Thomas, Laura Mayfield Tomokiyo, Takashi Tomokiyo, Christie Watson, Dorcas Wallace, and Boris Bartlog.

References

1. Black, A. and Taylor, P. and Caley, R.: 1998, "The Festival Speech Synthesis System", http://www.cstr.ed.ac.uk/projects/festival.html.
2. Black, A. W. and Taylor, P.: 1997, "The Festival Speech Synthesis System: system documentation", Human Communciation Research Centre, University of Edinburgh, Scotland, UK, HCRC/TR-83, January 1997, Avaliable at http://www.cstr.ed.ac.uk/projects/festival.html.
3. Carroll, John (ed.): 1996, Workshop on Robust Parsing, European Summer School in Logic, Language and Information (ESSLLI-96), Prague, Czech Republic, August 1996.
4. Danieli, Morena and Elisabetta Gerbino: 1995, 'Metrics for Evaluating Dialogue Strategies in a Spoken Language System', In *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pp. 34-39.
5. Dorr, Bonnie: 1992, 'Classification of Machine Translation Divergences and a Proposed Solution', *Computational Linguistics*.
6. Frederking, Robert, S. Nirenburg, D. Farwell, S. Helmreich, E. Hovy, K. Knight, S. Beale, C. Domashnev, D. Attardo, D. Grannes, and R. Brown: 1994, 'Integrating Translations from Multiple Sources within the Pangloss Mark III Machine Translation', *Proceedings of the First Conference of the Association for Machine Translation in the Americas* (AMTA-94). Columbia, Maryland.
7. R. Frederking, A. Rudnicky, and C. Hogan: 1997. Interactive Speech Translation in the DIPLOMAT Project. In Steven Krauwer et al., editors, *Proceedings of the Workshop on Spoken Language Translation*, pages 61–66, Madrid, Spain, July 1997. ACL/ELSNET.
8. Frederking, Robert, *et al.*: 1998, This issue.
9. Fritsch, Jürgen and Michael Finke: 1997, 'Improving Performance on Switchboard by combining Hybrid HME/HMM and Mixture of Gaussians Acoustic Models', *Proceedings of Eurospeech 97.*
10. Gates, Donna, Alon Lavie, Lori Levin, Alex Waibel, Marsal Gavaldà, Laura Mayfield, Monika Woszczyna, Puming Zhan: 1996, 'End-to-End Evaluation in JANUS: A Speech-to-Speech Translation System', *Proceedings of the ECAI 96*, Budapest.
11. Gavaldà, Marsal: 1998, The SOUP Home Page. `http://www.is.cs.cmu.edu/ISL.speech.parsing.soup.html`
12. Lavie, Alon: 1996, *GLR*: A Robust Grammar Focused Parser for Spontaneously Spoken Language* Ph.D. Thesis, Carnegie Mellon University.

13. Alon Lavie, Lori Levin, Yan Qu, Alex Waibel, Donna Gates, Marsal Gavalda, Laura Mayfield, Maite Taboada: 1996, Dialogue Processing in a Conversational Speech Translation System. Proceedings of the ICSLP 96, Philadelphia, USA, October 1996.
14. Lavie, Alon, Lori Levin, Puming Zhan, Maite Taboada, Donna Gates, Mirella Lapata, Cortis Clark, Matthew Broadhead, and Alex Waibel: 1997, 'Expanding the Domain of a Multi-lingual Speech-to-Speech Translation System', *Proceedings of the Workshop on Spoken Language Translation*, ACL/EACL-97, Madrid, Spain.
15. Levin, Lori and Sergei Nirenburg: 1994, 'The Correct Place of Lexical Semantics in Interlingual MT' COLING-94, Kyoto. pp. 349-355.
16. Levin, L., O. Glickman, Y. Qu, D. Gates, A. Lavie, C. Rosé, C. Van Ess-Dykema, and A. Waibel: 1995, "Using Context in Machine Translation of Spoken Language." In Proceedings of Theoretical and Methodological Issues in Machine Translation. Leuven, Belgium, 1995.
17. Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K.: 1992, *Machine Translation: A Knowledge-Based Approach*, Morgan Kaufmann.
18. Sergei Nirenburg (ed.). The Pangloss Mark III Machine Translation System. Technical report, Joint Technical Report, Computing Research Laboratory (New Mexico State University), Center for Machine Translation (Carnegie Mellon University), Information Sciences Institute (University of Southern California), 1995. Issued as CMU technical report CMU-CMT-95-145.
19. Yan Qu, Carolyn P. Rose, Barbara DiEugenio: 1996a, Using Discourse Predictions for Ambiguity Resolution. Proceedings of the COLING 96, Kopenhagen.
20. Qu, Y., DiEugenio, Lavie, Levin, Rosé: 1996b, "Minimizing Cumulative Error in Discourse Context." ECAI Workshop on Discourse. ECAI 1996.
21. Rogina, Ivica: 1997, Parameterraumoptimierung für Diktiersysteme mit unbeschränktem Vokabular PhD-Thesis, Fakultät für Informatik, Universität Karlsruhe.
22. Rosé,C., B. Di Eugenio, L. Levin, C. Van Ess-Dykema: 1995 "Discourse Processing of Dialogues with Multiple Threads." In Proceedings of the Association for Computational Linguistics. Cambridge, Massachusetts, 1995.
23. Tomita, Masaru: 1986, *Efficient Parsing for Natural Language*, Boston, Kluwer.
24. Thomas, Kavita: 1999, "Designing a Task-Based Evaluation Methodology for a Spoken Machine Translation System," Student Session of the ACL, 1999.
25. Waibel, Alex, Michael Finke, Donna Gates, Marsal Gavalda, Thomas Kemp, Alon Lavie, Lori Levin, Martin Maier, Laura Mayfield, Arthur McNair, Ivica Rogina, Kaori Shima, Tilo Sloboda, Monika Woszczyna, Torsten Zeppenfeld, Puming Zhan: 1996, 'JANUS-II Translation of Spontaneous Conversational Speech', *Proceedings of ICASSP 1996.*
26. Walker, Marilyn A., Diane J. Litman, Candace A. Kamm, and Alicia Abella: 1997, 'PARADISE: A Framework for Evaluating Spoken Dialogue Agents', *ACL/EACL 97*, pp. 271-280.
27. Ward, Wayne: 1990, 'The CMU Air Travel Information Service: Understanding spontaneous speech', In *Proceedings of the DARPA Speech and Language Workshop.*
28. Woszczyna, Monika: 1998, *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*, PhD-Thesis, Fakultät für Informatik, Universität Karlsruhe.

29. Woszczyna, Monika, N.Coccaro, A.Eisele, A.Lavie, A.McNair,T.Polzin, I.Rogina, C.P.Rose,T.Sloboda, M.Tomita, J.Tsutsumi, N.Aoki-Waibel, A.Waibel, W.Ward: 1993, 'Recent Advances in JANUS: a Speech Translation System', *Proceedings of Eurospeech, page 1295*.