
Machine Translation **15:** 3–25, 2000.

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
Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3891, USA
(E-mail: {lsl,alavie}@cs.cmu.edu)

Abstract. The JANUS-III system translates spoken languages in limited domains. The current research focus is on expanding beyond tasks involving a single limited semantic domain to significantly broader and richer domains. To achieve this goal, The MT components of our system have been engineered to build and manipulate multi-domain parse lattices that are based on modular grammars for multiple semantic domains. This approach yields solutions to several problems including multi-domain disambiguation, segmentation of spoken utterances into sentence units, modularity of system design, and re-use of earlier systems with incompatible output.

Key words: JRTk, multi-domain, robust parsing, semantic grammars, SOUP, speech translation

1. Introduction

Spoken Language Translation (SLT) systems have broken many barriers in the 1990s. Translation of well-formed, read speech with a small vocabulary (Woszczyna et al., 1993, 1994) 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. To achieve this goal, the MT components of our system have been engineered to build and manipulate multi-domain parse lattices that are based on modular grammars for multiple semantic domains. This approach yields solutions to several problems including efficient parsing and disambiguation in a large multi-domain search space, segmentation of spoken utterances into sentence units, modularizing system design, and re-using components with incompatible output.

⋆ All trademarks are hereby acknowledged.

The JANUS-III SLT translation system focuses on the broad domain of travel planning, scaling up from the JANUS-II domain of appointment scheduling (the 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 3,000 words in English and about 4,000 words in Spanish and German, which have more inflection, whereas 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 subdomains – for example, hotel accommodation, events and transportation – each of which has a number of subtopics 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 subdomains. 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, rewriting 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 possibly overlapping fragments of utterances. 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 (Lavie et al., 1996; Levin et al., 1995; Qu et al., 1996, 1997; Rosé et al., 1995).

Section 6 of the paper focuses on the engineering aspects of expanding our system to multiple domains. 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

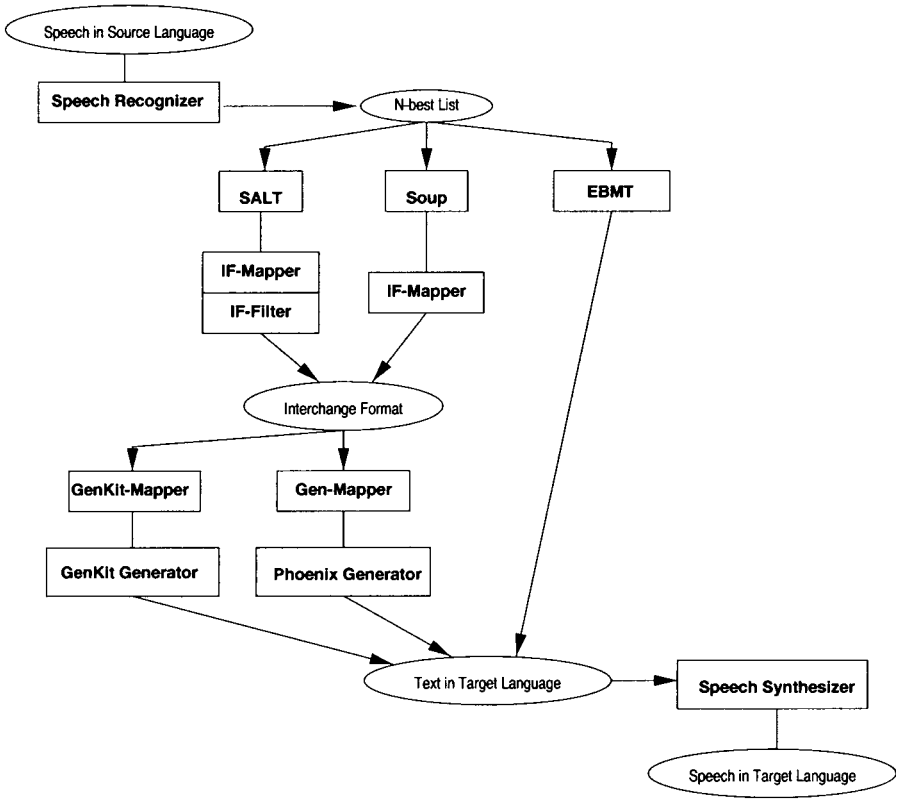

Figure 1. Components of the translation system.

main analysis component. Section 5 describes the interlingua representation we use for the travel planning domain. 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 SLT system for the travel domain can be seen in Figure 1. The main system modules are speech recognition, MT, 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 et al., 1999), a speech-synthesis system originally 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 MT (EBMT). This translation module was originally developed for the Pangloss and DIPLOMAT projects (Frederking et al. 1997, 2000; Nirenburg, 1995), and has been adapted for the travel domain. Experiments on how to combine the various translation approaches effectively are currently underway.

In the interlingua-based translation chain, translation is performed by first analyzing the source input string into an interlingua representation, and then generating a string in the target language from the interlingua. In our main analysis submodule, the input string is analyzed by SOUP (Gavaldà, 2000), 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 (IF) (see Section 5). 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 produces 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 user's with a mechanism for verifying analysis correctness, even when they are 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). We currently use two generation submodules. For English and Japanese, we use a rather simple semantic generator. 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. For German generation, we use the unification-based GenKit Generator (Tomita and Nyberg, 1988), which better supports morphological inflection via the Morphe morphology package. In this case, a generation mapper maps the IF into a feature structure appropriate for the GenKit grammar formalism.

The SOUP analyzer and the two generators 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.

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* can express availability for rooms, flights, and other travel-related facilities. 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 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 drawbacks. Semantic grammars are not easily adapted to new domains. Syntactic grammars on the other hand 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 (1). In Section 6 we describe how modular engineering of semantic grammars and our method for multi-domain integration provide for greater portability of semantic grammars. This should enable us to continue 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.

(1) a. The castle was built in the thirteenth century.
b. The temple has a beautiful garden.

3. Speech Recognition

For speech recognition in the JANUS 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 SLT task.

3.1. SPEECH RECOGNITION COMPONENTS

The goal of speech recognition for SLT 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 defined in (2) is largest.

$$P(W|A) = \frac{P(A|W)P(W)}{P(A)} \tag{2}$$

Here, $P(A)$ is the probability of observing the recorded acoustic 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 the "acoustic model".

It is important to understand that a considerable number of simplifications and approximations are required to make this maximization problem tractable 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 from 5 (for simple tasks) to 50 (for fast, spontaneous telephone speech with strong coarticulation).

3.2. 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 is 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 used to map between words and their corresponding sequence of phonemes. Figure 2 shows a few examples taken from our 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 the 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.

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

Figure 2. Examples taken from a JRTk pronunciation dictionary.

Since the sound of a phoneme varies depending on the adjacent phonemes, the phonemes are commonly subdivided into three segments that are modeled depending on the identity 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 SLT system have only 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.

3.3. LANGUAGE MODELS

The language model is used to compute the *a priori* likelihood for a word sequence based on statistical knowledge derived from transcribed dialogues and related texts. The JRTk-based recognizer in our system uses a trigram language model 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 parser was inspired by Wayne Ward's (1990) Phoenix parser and is a robust 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 3 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 1. 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.

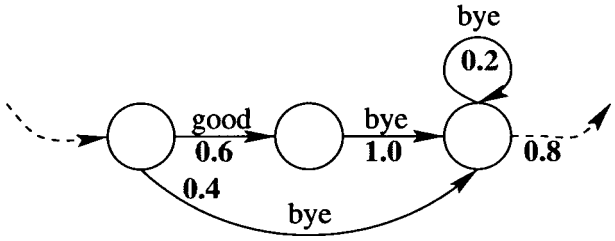

Figure 3. PRTN for the RHS sequence `( *good +bye )`.

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:

Maximize coverage: Given two interpretations, prefer the one that covers the highest number of input tokens.

Minimize number of parse-trees: Given two interpretations, prefer the one that has fewer parse-trees. The rational behind this principle is to try to minimize parse fragmentation.

Minimize the number of parse tree nodes: Given two interpretations, prefer the one that has fewer parse tree nodes.

Minimize the number of wildcard matches: Given two interpretations, prefer the one that has fewer usages of the wildcard symbol (_any_).

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.

Maximize the 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 (see Section 6), 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 W, i.e. to maximize the probability $\mathrm{P}(T|W)$, where $T = (t_1, t_2, \cdots, t_k)$ is the sequence of subdomains corresponding to the sequence of parse trees in the interpretation.

$$\mathrm{P}(T|W) = \frac{\mathrm{P}(W|T) \cdot \mathrm{P}(T)}{\mathrm{P}(W)} \tag{3}$$

Since $P(W)$ does not change once the utterance has been recognized, this is the same as maximizing $P(W|T) \cdot P(T)$. To simplify the computation, we assume that the probability of the words W_i covered by parse tree i depends only on the domain t_i to which the parse tree belongs. Thus,

$$P(W|T) = \prod_i P(W_i|t_i) \tag{4}$$

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 $P(T)$ within one utterance is approximated by a unigram or a bigram statistic (5).

$$\begin{aligned} P(T) &\approx P(t_1) \cdot P(t_2) \cdot \ldots \cdot P(t_N) \\ &\approx P(t_1) \cdot P(t_2|t_1) \cdot \ldots \cdot P(t_N|t_{N-1}) \end{aligned} \tag{5}$$

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 5,000 non-terminals, 18,000 rules, and 8,000 lexical entries, giving rise to a collection of PRTNs in the order of 39,000 nodes and 73,000 arcs.

For more efficient grammar development, we have constructed a graphical grammar editor, called G-SOUP, that allows for

(a) graphical visualization, creation, deletion and editing of non-terminals and rules;
(b) automatic assessment of rule coverage;
(c) automatic detection of rule conflicts; and
(d) automatic and manual annotation of rules.

In its performance, SOUP, which is implemented in C++, is very efficient. On an English grammar for the scheduling task, containing 600 concepts (21 top-level, 466 auxiliary), 2,880 grammar rules and 829 lexical entries, which give rise to 6,373 grammar nodes and 10,480 grammar arcs, running on a Sun-Ultra-I at 167 MHz, a set of 609 sentences containing a total of 5,502 words were parsed in 4.352 seconds, 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 multilingual translation in the context of the C-STAR consortium (Levin et al., 1998). Interlingual MT 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 that has all-ways translation with existing languages requires only writing one analyzer that maps utterances into the interlingua and one generator that maps interlingua representations into sentences. A consequence of this 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 their own 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, 1994; Levin & 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 flight. These DAs form the basis of the C-STAR interlingua, the 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, (6) uttered by the customer and (7) uttered by the agent are both requests for information about credit cards as a form of payment.

(6) Do you take credit cards?

(7) Will you be paying with a credit card?

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 (8). However, there are constraints on the order and combination of concepts so that not all sequences of concepts are possible. The

current IF definition consists of 55 speech acts, 84 concepts, and 119 argument types. Argument values are less restricted and number in the thousands.

(8) `speaker: speech act +concept* argument*`

In (9) 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 (10) 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, (11) demonstrates a DA which contains neither concepts nor arguments.

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

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

(11) 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 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 exploited. For large domains, this can result in very cumbersome grammars that become difficult to develop further, and which are highly ambiguous. 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 similarly to divide the task of grammar development into well defined subgrammars that can be developed and maintained independently, while sharing common subgrammar 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 sudomains in a fairly straightforward way. Each of the subdomain grammars builds upon lower-level concepts, some of which are likely to appear in more than one subdomain (e.g., 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 subdomain grammars.

In our system, we divided the large travel planning domain into four main subdomains: 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 subdomains). 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 subdomain grammars. These include time and date expressions (such as *around 5pm on Friday*) as well as lists of proper names (e.g., *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 subgrammar substantially reduce the effort required to expand the system to new domains.

6.2. ANALYSIS WITH MULTIPLE-DOMAIN GRAMMARS

In parallel to the modular design of subdomain grammars and shared grammar files, the Soup parser was extended in order to allow it to support the integration of several domain grammars and interlingua representations in an elegant and efficient way. As in the case of a single-domain system, the task of the parser is to analyze a spoken input utterance as a sequence of top-level concepts, which are the root nodes of the subdomain grammars. 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 subdomain grammars must be considered during parse time. Working with several 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 top-level concepts drawn from the various domain grammars.

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 parse effectively with multiple grammars concurrently. In effect, the parser works with a large "union" grammar that consists of the separate domain grammars tied together at the root of the grammar. 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, non-terminals 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.

The SOUP parser was designed for spoken language, which is disfluent and does not reflect sentence boundaries. Each utterance is therefore not parsed as a single tree dominated by a single root node, but as a sequence of top-level concepts, possibly interspersed with unparsable input segments. In our grammars, top-level concepts correspond to speech acts such as informing, requesting, and acknowledging. Thus SOUP's ability to analyze an utterance as a sequence of concepts eliminates the need for 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. An example of such a lattice can be seen in Figure 4. Only the parsable top-level concept labels are shown in the figure. Note that each concept is annotated with a label corresponding to the domain grammar from which it originated. 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. One component of the scoring heuristic is statistical domain information, as described in Section 4.3. Note that the concepts that comprise one utterance do not all have to originate from the same subgrammar. The utterance in (12) contains concepts (and subparses) from three different subdomain grammars.

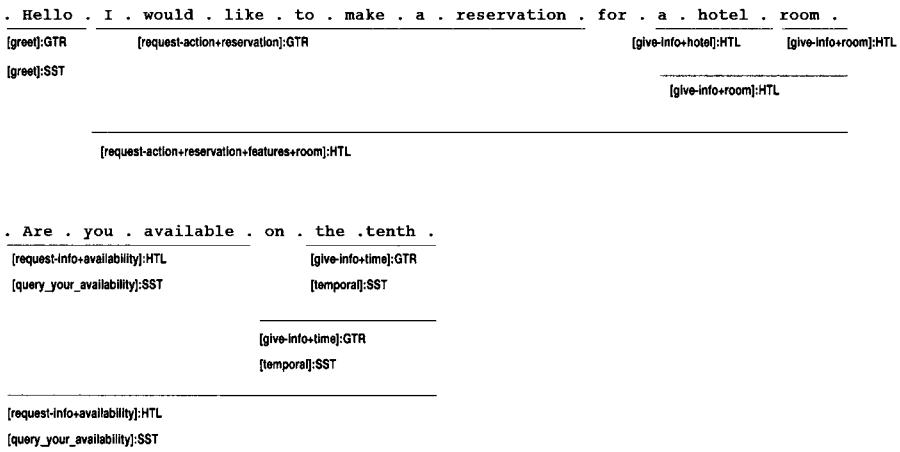

Figure 4. A portion of a parse lattice produced by the SOUP Parser.

(12) 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, embedded within the General Travel grammar, while the words *Frankfurt* and *on the fifth* are parsed by the shared grammar, in this case accessed by the subdomain grammar for transportation.

A considerable advantage of our approach is that grammars producing different interlingua representations can be integrated into one system on the subutterance 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 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 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 (13), the utterance is analyzed into two top-level concepts and interlingua representations, the first from the Hotel subdomain (using an IF interlingua), while the second is from the scheduling domain (using its own interlingua representation).

(13) I would like to make a reservation for a hotel room – are you available on the tenth?

```
(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 subdomain grammars, and how the multiple domains are handled by our analysis module.

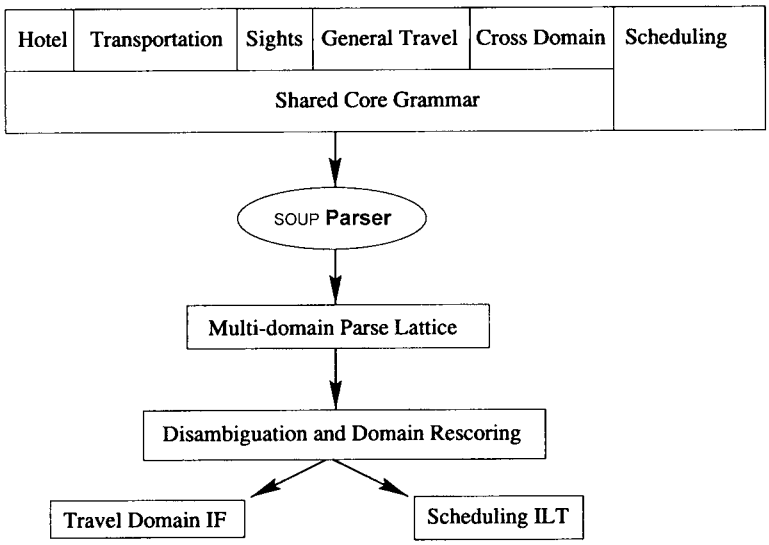

Figure 5. Combining multiple subdomain grammars with shared and cross-domain grammars.

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 for evaluation of 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 the human–computer interface aspects of the system.

The subjects involved in all user studies had little or no previous exposure to speech recognition or SLT. They were seated in a moderately noisy office and asked to play the role of a traveler booking a trip to Germany or, in the case of the third 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 SLT system, translating from English via IF to English; our multimodal interface allowing for handwriting recognition and sharing web-pages; and a muted netmeeting video-conference (no audio).

During the entire duration of the user study, the subjects were minimally supervised and videotaped for later analysis. Instructions on how best to use the system and interventions in case of problems were kept to a minimum.

The data collected in the user studies was first used for system evaluation. We conducted sentence-level evaluations of the entire end-to-end translation system, from speech input all the way to translation output. Although the target language for the user studies was English, we also produced Japanese and German for the purpose of evaluation. Bilingual graders compared the source-language input and target-language output for each sentence. The grades assigned were "perfect", "OK", and "bad". "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. (1997). 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 a subject trying to book a trip to Japan.

Table I. Translation grades for English–English, English–Japanese, and English–German translation using the SOUP parser

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

Experiment 1 in Table I shows an evaluation of the quality of the output produced by the speech recognizer, measured by the same criteria we use for evaluating the output of the translation engine: a grade of "OK" for retaining all relevant meaning and a grade of "Perfect" for being fluent. For about 22% of all utterances, some important change of meaning 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 from 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 were not translated correctly due to the lack of coverage of the interchange format or the grammars.

Experiments 4 and 5 give the performance for English–Japanese translation from transcribed English input (Experiment 4) and recognized English input (Experiment 5). The slightly better results in comparison to English–English paraphrase reflect the subjective nature of the grading process more than any real difference in performance. Experiments 6 and 7 report the numbers for English–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 four months), resulting in lower coverage. Table II shows the progress of the grammar development for English–English translation over a recent six-month period.

Table II. Translation grades for different development stages, English–English translation

Date	Method	OK+Perfect (%)	Perfect (%)
January 1999	Transcription	69	46
January 1999	Recognition	55	36
April 1999	Transcription	70	49
April 1999	Recognition	57	38
August 1999	Transcription	74	54
August 1999	Recognition	59	42

Table III reports the results for English–German translation using the Pangloss EBMT direct translation approach. While the results look encouraging, the low percentage of perfect translations reflects the fact that it is difficult to get high-quality German output using the EBMT method. However, this approach offers an excellent fall-back strategy for uncovered or out-of-domain utterances.

Table III. Translation grades for the English–German EBMT

	Method	Output language	OK+Perfect (%)	Perfect (%)
8	Transcription	German	80	36
9	Recognition	German	67	31

At a first glance, the sentence-based evaluation results may seem rather disappointing. However, further analysis indicates that 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 20-sentence dialogue being translated completely correctly is less than 1%. However, this 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 primary dialogue goals, namely to book their flight and a hotel room, as well as to get some information 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 successfully. Some secondary dialogue goals, such as getting directions to a sushi restaurant near the hotel or obtaining a map of the train station correspond to concepts that fall beyond the coverage of the various system components (speech recognition, grammars, IF, agent databases), and were therefore impossible to achieve. We are still working on a full task-based evaluation (Thomas, 1999) 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 architecture of the JANUS MT engine described in this paper has provided a solid design foundation for our translation system for the travel domain. Much of our current work involves incremental improvements in the coverage of our grammars and other knowledge sources as well as adding new languages. We are also working however on a number of advanced extensions to the translation system itself. These include more advanced statistical disambiguation techniques and the development of several alternative translation methods that we intend to combine with our grammar-based approach.

8.1. MULTI-ENGINE TRANSLATION

Multi-engine translation was proposed by Frederking & Nirenburg (1994) and has since been implemented in the Diplomat (Frederking et al., 1997, 2000) and Verbmobil (Ruland et al., 1998) 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, we are working to combine the three translation chains shown in Figure 1: the rule-based parsing system, the statistical parsing system, and the EBMT direct translation approach. A major research issue in multi-engine translation is improving methods for combining the outputs of the various engines (Frederking et al., 2000).

8.2. 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.

8.3. INTEGRATION OF SLT WITH MULTIMODAL INTERFACES

One possible usage scenario for our travel-domain SLT system involves a videoconference between a travel agent in a foreign country and an interested client who does not speak the language of the agent. For this type of scenario, a vast array of additional tools for communication can greatly complement the SLT itself. The travel agent may wish to 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 travel-information databases through the same interface that is used for the communication with the client, especially if the SLT involved in this communication already provides speech understanding components that can also be used for database access. Our current C-STAR demonstration system already integrates a number of multimedia techniques such as speech, handwriting, and gesture recognition, and face tracking in a common human–computer–human interface. We plan to continue to work on advanced issues related to multimodal integration, such as deictic references.

8.4. TASK-BASED EVALUATION

Our current sentence-level evaluations measure the accuracy of translation, but do not show whether or not 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 (Danieli & Gerbino, 1995; Walker et al., 1997) but less frequently to human–human dialogue mediated by machine. 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.

8.5. INTERNATIONAL COOPERATION

The IF 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 other C-STAR partners was run through the English and Japanese generators at CMU and vice versa. The IF was then modified to accommodate problems identified. Using a specially designed C-STAR communication protocol, we have successfully integrated systems from all six C-STAR partners, each running their own translation system locally, and communicating (primarily via IF) with the other partner systems in order to achieve close to real-time translation between the six C-STAR languages. A large international demonstration to the public and the media was conducted by all six C-STAR partners on July 21, 1999, with great success. Details of the demonstration (including video clips) can be found on the C-STAR web site at `http://www.c-star.org`.

Acknowledgements

The IF formalism is the result of a close cooperation of the six C-STAR-II partners. 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 Alexandra Slavkovic. Part of the work on statistical DA identification was done by Dr. 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 Boris Bartlog, Daniela Müller, Kavita Thomas, Laura Mayfield Tomokiyo, Takashi Tomokiyo, Dorcas Wallace, Taro Watanabe, and Christie Watson.

References

Black, Alan W., Paul Taylor and Richard Caley: 1999, *The Festival Speech Synthesis System: System Documentation*, Human Communciation Research Centre, University of Edinburgh, Scotland; avaliable at `http://www.cstr.ed.ac.uk/projects/festival/manual`. 17th June 1999.

Danieli, Morena and Elisabetta Gerbino: 1995, 'Metrics for Evaluating Dialogue Strategies in a Spoken Language System', *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford, California, pp. 34–39.

Dorr, Bonnie J.: 1994. 'Machine Translation Divergences: A Formal Description and Proposed Solution', *Computational Linguistics* **20**, 597–633.

Frederking, Robert and Sergei Nirenberg: 1994, 'Three Heads are Better than One', *4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, pp. 95–100.

Frederking, Robert, Alexander Rudnicky and Christopher Hogan: 1997, 'Interactive Speech Translation in the DIPLOMAT Project', *Spoken Language Translation: Proceedings of a Workshop Sponsored by the Association for Computational Linguistics and by the European Network in Language and Speech (ELSNET)*, Madrid, Spain, pp. 61–66.

Frederking, Robert, Alexander Rudnicky, Christopher Hogan and Kevin Lenzo: 2000, 'Interactive Speech Translation in the DIPLOMAT Project', this volume.

Gates, Donna, Alon Lavie, Lori Levin, Marsal Gavaldà, Monika Woszczyna and Puming Zhan: 1997, 'End-to-End Evaluation in JANUS: a Speech-to-Speech Translation System', in E. Maier, M. Mast and S. Luperfoy (eds), *Dialogue Processing in Spoken Language Systems*, Berlin, Springer Verlag, pp. 195–206.

Gavaldà, Marsal: 2000, 'A Parser for Real-world Spontaneous Speech', *IWPT 2000: Sixth International Workshop on Parsing Technologies*, Trento, Italy. Paper available (28 Feb 2001) at `http://www.cs.cmu.edu/~marsal/papers/iwpt2000.html`.

Lavie, Alon, Lori Levin, Yan Qu, Alex Waibel, Donna Gates, Marsal Gavaldà, Laura Mayfield and Maite Taboada: 1996, 'Dialogue Processing in a Conversational Speech Translation System', *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP-96)*, Philadelphia, PA, pp. 554–557.

Levin, Lori, Donna Gates, Alon Lavie and Alex Waibel: 1998, 'An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues', *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, Sydney, Australia, pp. 1155–1158.

Levin, Lori, Oren Glickman, Yan Qu, Donna Gates, Alon Lavie, Carolyn P. Rosé, Carol Van Ess-Dykema and Alex Waibel: 1995, 'Using Context in Machine Translation of Spoken Language', *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, TMI 95*, Leuven, Belgium, pp. 173–187.

Levin, Lori and Sergei Nirenburg: 1994, 'The Correct Place of Lexical Semantics in Interlingual MT', *COLING 94: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 349–355.

Nirenburg, Sergei (ed.): 1995, *The Pangloss Mark III Machine Translation System*, Joint Technical Report CMU-CMT-95-145, Computing Research Laboratory (New Mexico State University, Las Cruces, NM), Center for Machine Translation (Carnegie Mellon University, Pittsburgh, PA), Information Sciences Institute (University of Southern California, Marina del Rey, CA).

Nirenburg, Sergei, Jaime Carbonell, Masaru Tomita and Kenneth Goodman: 1992, *Machine Translation: A Knowledge-Based Approach*. San Mateo, California: Morgan Kaufmann.

Qu, Yan, Barbara Di Eugenio, Alon Lavie, Lori Levin and Carolyn P. Rosé: 1997, 'Minimizing Cumulative Error in Discourse Context', in E. Maier, M. Mast and S. Luperfoy (eds), *Dialogue Processing in Spoken Language Systems*, Berlin, Springer Verlag, pp. 171–182.

Qu, Yan, Carolyn P. Rosé and Barbara Di Eugenio: 1996, 'Using Discourse Predictions for Ambiguity Resolution', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 359–363.

Rosé, Carolyn Penstein, Barbara Di Eugenio, Lori S. Levin and Carol Van Ess-Dykema: 1995, 'Discourse Processing of Dialogues with Multiple Threads', *33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, pp. 31–38.

Ruland, Tobias, C. J. Rupp, Jörg Spilker, Hans Weber and Karsten L. Worm: 1998, *Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language*, also *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, Sydney, Australia, pp. 1167–1170.

Thomas, Kavita: 1999, 'Designing a Task-Based Evaluation Methodology for a Spoken Machine Translation System', *37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland, pp. 569–572.

Tomita, Masaru: 1986, *Efficient Parsing for Natural Language*, Boston, MA: Kluwer.

Tomita, Masaru and Eric H. Nyberg: 1988, *Generation Kit and Transformation Kit, Version 3.2: User's Manual*, Technical Report CMU-CMT-88-MEMO, Carnegie Mellon University, Pittsburgh, PA.

Walker, Marilyn, D. Litman, C. Kamm and A. Abella: 1997, 'PARADISE: A Framework for Evaluating Spoken Dialogue Agents', *35th Annual Meeting of the Association for Computational*

Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, pp. 271–280.

Ward, Wayne: 1990, 'The CMU Air Travel Information Service: Understanding Spontaneous Speech', *DARPA Workshop on Speech and Natural Language Processing*, Hidden Valley, PA.

Woszczyna, Monika: 1998, *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*, Ph.D. thesis, Fakultät für Informatik, Universität Karlsruhe, Germany.

Woszczyna, Monika, N. Aoki-Waibel, Finn Dag Buø, Noah B. Coccaro, K. Horiguchi, Thomas Kemp, Alon Lavie, Arthur McNair, Thomas S. Polzin, Ivica Rogina, C. P. Rosé, Tanja Schultz, B. Suhm, M. Tomita and Alex Waibel: 1994, 'Janus 93: Towards Spontaneous Speech Translation', *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, Adelaide, Australia, Vol. 1, pp. 345–348.

Woszczyna, Monika, N. Coccaro, A. Eisele, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. P. Rosé, T. Sloboda, M. Tomita, J. Tsutsumi, N. Aoki-Waibel, A. Waibel and W. Ward: 1993, 'Recent Advances in JANUS: a Speech Translation System', *Eurospeech: Proceedings of the 3rd European Conference on Speech, Communication and Technology*, Berlin, Germany, pp. 1295–1298.