Computer Speech and Language (1999) **13**, 177–194
Article No. csla.1999.0119
Available online at http://www.idealibrary.com on IDEAL

Stochastically-based semantic analysis for machine translation

Wolfgang Minker,∗† Marsal Gavaldà‡ and Alex Waibel§‖

†Spoken Language Processing Group, LIMSI-CNRS, BP. 133, 91403 Orsay Cedex, France ‡Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A. §Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15221, U.S.A. and ‖Universität Karlsruhe, Fakultät für Informatik, 76131 Karlsruhe, Germany

Abstract

We report our experience of applying a stochastic method for understanding natural language to a multilingual appointment scheduling task, in particular, to the English spontaneous speech task (ESST). The aim of the spoken language systems developed for this task is to translate spontaneous conversational speech among different languages. We have investigated the portability of a stochastic semantic analyser from a setting of human–machine interactions air travel information services (ATIS) and multimodal multimedia automated service kiosk (MASK) into the more open one of human-to-human interactions (ESST).

1. Introduction

In this article, a stochastic component for natural language understanding, initially developed as a part of a spoken language system for the information retrieval applications air travel information services (ATIS) (Minker, Bennacef & Gauvain, 1996) and multimodal multimedia automated service kiosk (MASK) (Minker, 1997), is ported to a multilingual, appointment scheduling task, the English spontaneous scheduling task (ESST).

Machine translation systems combine various component technologies (such as speech recognition, natural language understanding) to provide understanding of the meaning of a spoken utterance. Additionally, natural language generation and speech synthesis are used to build end-to-end systems which accomplish a given task, such as the scheduling of an appointment by interlocutors speaking different languages. In the context of such systems, today's state-of-the-art rule-based methods for natural language understanding provide good performance in limited applications. However, the *manual* development of an understanding component is costly, as each new application, task, or domain requires its own adaptation or, in the worst case, a completely new implementation. In this study, statistical modeling techniques are used, first, to model the semantic content of naturally-occurring sentences, and then, to replace the commonly-used hand-generated grammar rules that parse the recognizer output into

[∗] Author for correspondence: E-mail: `minker@limsi.fr`

0885–2308/99/020177 + 18 $30.00/0

a semantic representation. The statistical models are derived from the *automatic* analyses of large corpora of naturally-occurring sentences along with their semantic representations. This facilitates the porting to different tasks, as well as to new languages. Such stochastic methods have been applied in the BBN-HUM (Schwartz, Miller, Stallard & Makhoul, 1996) and the AT&T-CHRONUS (Levin & Pieraccini, 1995) systems for the American ARPA-ATIS task. To date, the language and domain portability of stochastic parsers has not been investigated. However, this represents one of the essential arguments for applying a stochastic method for the semantic analysis.

In the work reported in this article, the portability of a stochastically trained grammar from a setting of human–machine interaction, ATIS and MASK, to one of human-to-human interaction is investigated. The stochastic component has been trained using a corpus annotated by the CMU-PHOENIX parser, which, as part of the JANUS speech-to-speech translation system, transforms the output of the speech recognizer into semantic trees. In contrast to parsing spontaneous human–machine queries, spoken-language translation requires the analysis of a human-to-human activity. This challenging problem is being explored at various laboratories and by national and international research programs (see, for example, the C-STAR international Consortium for Speech Translation Advanced Research, or the German VERBMOBIL (Bub & Schwinn, 1996) project). Since a translation system deals with human-to-human dialogs, as opposed to the ATIS and MASK tasks in which a person negotiates with a machine, not only the domains per se, but also the behavior of the interlocutors differ greatly, especially with regard to negotiation patterns and degree of spontaneity. Such a system therefore requires not only robustness to spoken language phenomena (i.e. false starts, filler words, ungrammatical constructions, and other disfluencies and inaccuracies) but also a higher representational resolution than the one needed for human–machine tasks.

In the following sections, we describe how both the rule-based (PHOENIX) and the stochastic parsers work (Sections 2 and 3), and introduce the stochastic model (Section 4). In Section 5, the semantic representation is defined, and the main characteristics of the training corpus are provided in Section 6. Section 7 discusses comparative evaluations between the stochastic component and PHOENIX.

2. Rule-based parsing in PHOENIX

The PHOENIX parser was first used for the ATIS task (Ward & Issar, 1995) but, as depicted in Figure 1, it was adopted as one of the parsing engines of the JANUS speech-to-speech translation system (Lavie *et al.*, 1997). The results reported in this article arise from a particular encarnation of JANUS to handle the ESST domain.

The JANUS speech recognizer is based on the JANUS Recognition Toolkit (Finke *et al.*, 1997). It uses streams of input features derived from Mel-scale, cepstral or PLP filters processed via linear discriminant analysis. The acoustic units are context-dependent, modeled via continuous-density HMMs. The trigram language model is built from ESST data and other corpora.

The PHOENIX translation modules used in JANUS (Ward, 1994, Mayfield, Gavaldà, Ward & Waibel, 1995) consist of a top–down chart parser that, given an analysis grammar for the source language and an input sentence in that language, produces a semantic tree, and of a simple generation module that, given a generation grammar for the target language and a semantic tree, produces a surface form of the semantic tree in the target language. The parser uses heuristics to, in this order, maximize coverage and minimize tree complexity.

PHOENIX grammars are context-free grammars (CFG) in which the left-hand sides (rule

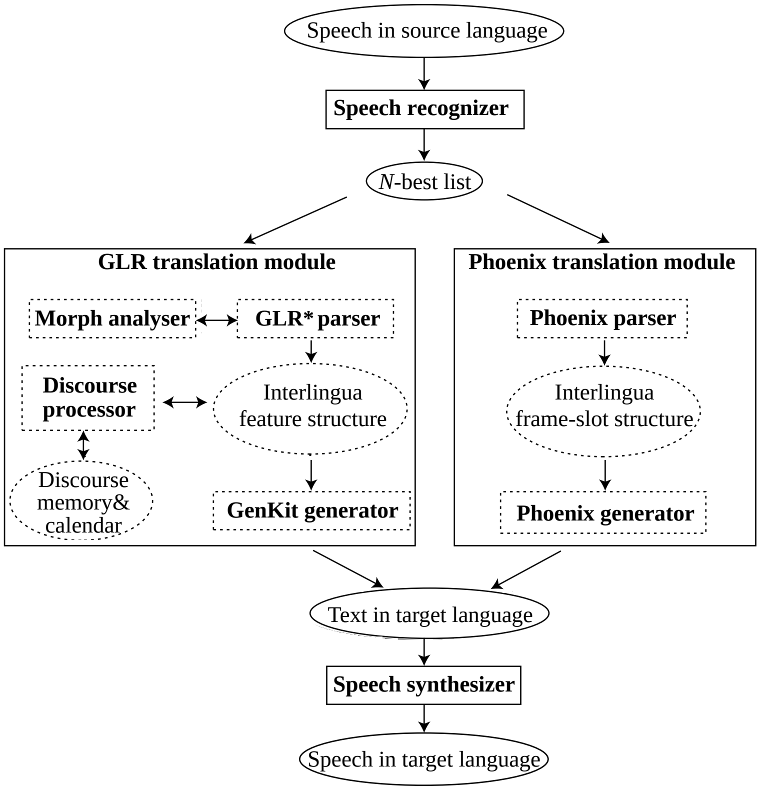


Figure 1. JANUS system diagram (from Lavie *et al.*, 1997).

heads) correspond to semantic tokens relevant to the application at hand, and right-hand sides (rule bodies) capture a particular way in which the semantic token can be expressed. Figure 2 shows some sample grammar rules for a scheduling domain. Note that grammar terminals (i.e. lexical items) and grammar non-terminals (i.e. semantic tokens) are freely mixed in the right-hand sides. Auxiliary non-terminals (upper cased in the example) are used only as a shorthand for the grammar writer and do not appear in the final parse tree.

There are two stages in the development of a semantic grammar. First, the relevant concepts of the domain have to be established. This corresponds to finding the non-terminals of the grammar. Then, in an arduous and lengthy process, appropriate right-hand sides need to be written to try to capture all the possible ways in which a particular concept can be expressed. Around 300 man-hours over the course of a year were employed to develop the PHOENIX ESST grammar, consisting of 600 non-terminals (of which 21 are top-level), 2880 rules and 831 terminals.

Once the grammar is deemed developed enough, it is compiled by PHOENIX into Recursive Transition Networks (RTN), each grammar non-terminal giving rise to one RTN. A subset of the non-terminals are marked as starting symbols of the grammar, i.e. able to stand at the root of a parse tree. Also, skipping of input words is only allowed between them.

Given an input sentence to be parsed, PHOENIX pre-processes it by eliminating out-of-vocabulary words (i.e. words not appearing in the grammar) and expanding some contractions

```
(1)  [farewell]
                        ( *good +bye )
(2)  [suggest_meeting]
                        ( SUGGESTION MEET *[time] *[location] )
                        ( is MEET GOOD *FOR_YOU )
                        ...
(3)  SUGGESTION
                        ( how about )
                        ( what *do *you *think about )
                        ...
(4)  MEET
                        ( *if *we meet )
                        ( meeting )
                        ...
 ...
```

Figure 2. Sample grammar rules for a scheduling task. Lexical items (in lower case) and calls to semantic nets (in upper case or enclosed in square parentheses) are freely mixed. A "*" indicates optional token, a "+" indicates repeatible token, a "*+" is equivalent to the Kleene star, i.e. indicates that token can occur zero or more times. For instance, rule (1) accepts *bye*, *goodbye*, *bye bye*, etc.

(e.g. *I'm* → *I am*). Then, the parse engine conducts a left-to-right Viterbi search in which all possible traversals of the RTNs are pursued (top–down) as long as they match the input words. Pruning and scoring heuristics include maximizing coverage, i.e. prefer those interpretations (semantic trees) that cover the largest number of input words, and minimizing tree complexity, i.e. prefer those interpretations that contain the smallest number of subconcepts (nodes of the semantic tree). The resulting top-ranked interpretation is the most coherent semantic tree of the input utterance, according to the given grammar.

3. Stochastically-based semantic analysis

The semantic analyser using a stochastic method is based on the theory of hidden Markov models (HMM). The functional diagram is given in Figure 3. We distinguish two processing steps: during *training*, the parameter estimator establishes the model from the transcribed utterances (output of the speech recognition component) and their corresponding semantic representations. In the *decoding* or *testing* stage, the semantic decoder, implemented as an ergodic HMM (Rabiner & Juang, 1986), outputs the most likely semantic representation given a transcribed utterance as input.

The stochastic component uses the same techniques for training and decoding that were developed for ATIS and MASK, thus achieving a certain degree of portability and flexibility. Only the data sets and their encoding are domain specific. In our new ESST task, the semantic sequences used for training and evaluation are derived from the parse trees that were automatically produced by the CMU-PHOENIX parser. Using these annotations and an appropriate paradigm for evaluating translation accuracy provided the means to validate the stochastic component and to compare it with the rule-based method.

Figure 4 describes the porting. In the pre-processing step of data segmentation, the tran-

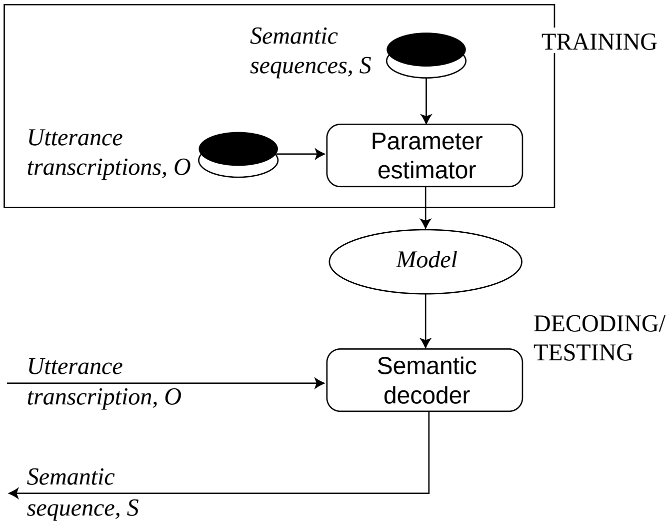


Figure 3. Overview of the semantic analyser of a spoken language system using a stochastic method (Minker, Bennacef & Gauvain, 1996).

scriptions of the utterances were broken down into smaller semantic discourse units (SDUs) using a combination of acoustic, lexical, semantic and statistical knowledge sources, as described by Lavie, Gates, Coccaro and Levin (1996). As a part of the stochastic component, the treetoexpr module converts the semantic PHOENIX concept-parses, provided as semantic trees, into a sequential representation. The parameter estimator then establishes a HMM using the SDUs and the corresponding semantic representation. Given a test utterance, the semantic decoder provides a semantic hypothesis (sequence which is reconverted by the exprtotree interface into a tree-based representation). This representation is then used by the PHOENIX generator to produce the translated utterance in the target language.

4. Knowledge representation

The parameters of the stochastic model are estimated given sequences of words and their corresponding semantic representation.

4.1. Semantic representation

The semantic tree representations produced by PHOENIX [Fig. 5(a)] are similar to those applied by BBN in the HUM system (Schwartz *et al.*, 1996), except that PHOENIX trees use only semantic labels, whereas HUM trees are semantic and syntactic. The PHOENIX parses rely on a case grammar formalism (Bruce, 1975). The parser tries to model the relevant ESST information structures as well as the lexical realization of these structures in various languages. Several *semantic tokens* correspond to concepts and subconcepts in an utterance. Based on transcripts of English and German scheduling dialogs, a set of fundamental semantic tokens was defined to represent the relevant concepts the speakers use for this task. These semantic tokens can be seen as the vertices of a directed acyclic graph in which the edges correspond to concept–subconcept relations. Table I shows the semantic tokens used for ESST. They are roughly listed in an abstract-to-concrete ordering. Typical high-level (more

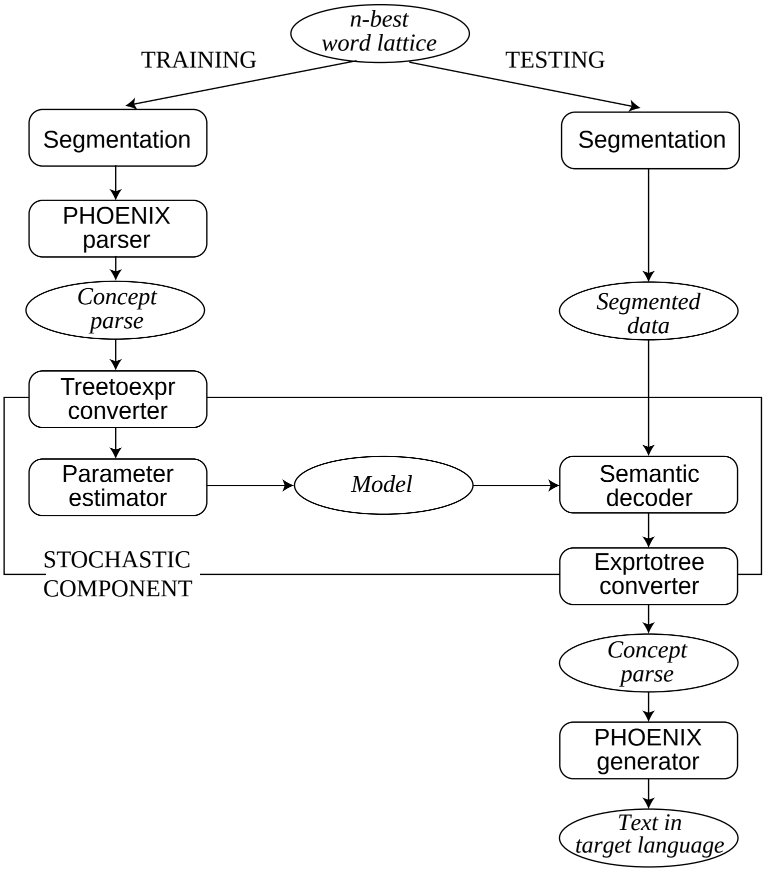


Figure 4. Porting of the stochastic component for natural language understanding to the ESST domain.

general) tokens (*L1*) are <agree>, <interject>, <give_info> and <temporal>; examples of lowest-level (more specific) tokens (*L4*) are <hour>, <minute> and <year>. The tokens are combined so as to build a tree-based meaning representation. For example, in Figure 5(a), a typical temporal concept <temporal> has <point> and <interval> as daughter concepts, and <interval> in turn has daughters <start_point> and <end_point>, etc. The leaves of the tree correspond to the lexical items present in the input utterance, e.g. *probably*, *sometime*, etc. Each speech-act contains a separate top-level concept (root of a semantic tree). The speech-acts are then concatenated without any ordering constraints. The example utterance is parsed into the independent semantic speech-acts <interj>, <temporal>, <agree> that capture the top-level meaning of *probably*, *sometime between nine and five* and *would be good*.

The rule-based PHOENIX output is not in a form which can be directly used by the model parameter estimator of the stochastic component. In order to estimate these parameters, each word of the input utterance must have a corresponding *semantic label*. The concept parses were adapted by treetoexpr (Fig. 4). This module converts the tree-based representation [Fig. 5(a)] into sequences of semantic *tree-labels* [Fig. 5(b)]. Each tree-label represents the complete path from the root down to the lowest level token. An example path trough the tree is <temporal>–<interval>–<start_point>–<time>–<hour>↦*nine*. This exhaustive and deep semantic rep-

TABLE I. Semantic tokens used for the scheduling task, categorized by degree of abstraction. An example path through the tree would be “<temporal>–<interval>–<start_point>–<time>–<hour>–*nine*” (cf. Fig. 5)

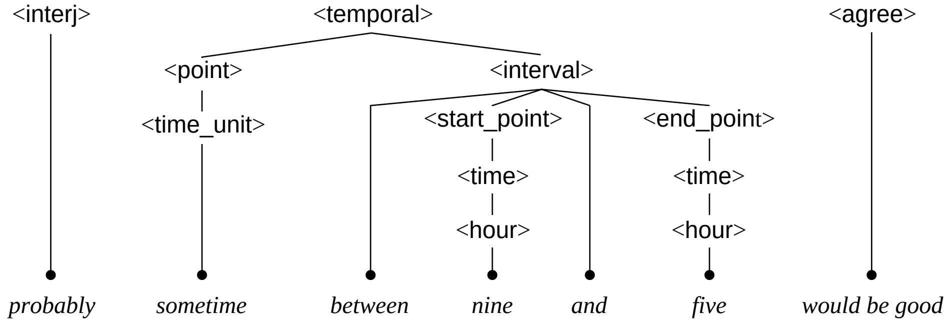
L1	<agree>, <conditional>, <confirm>, <correction>, <give_info>, <interj>, <move>, <nicety>, <no>, <q_your_availability>, <q_your_knowledge>, <reject>, <request_clarification>, <request_confirmation>, <sugg_loc>, <sugg_meet>, <sugg_time>, <**temporal**>, <yes>, <your_turn>
L2	<any_other_time>, <anytime>, <available_again>, <better_temp>, <conj>, <cycle>, <day_of_week>, <date>, <dur_num>, <duration>, <farewell>, <first>, <greeting>, <how_long>, <i_sugg>, <if_clause>, <in_next_couple_weeks>, <**interval**>, <is_that_okay>, <lets_consider>, <lets_do_x>, <loc_name>, <loc>, <mealtime>, <month_name>, <my_availability>, <my_preference>, <my_reluctance>, <my_unavailability>, <name>, <neg_babble>, <not_enought_time>, <only>, <other>, <out_of_town>, <please_wait>, <point>, <range>, <time_unit>, <thanks>, <then_clause>, <then>, <though>, <todays_date>, <too>, <two_hour_block>, <unit>, <when>, <where>, <within>, <worse_temp>, <your_availability>, <your_unavailability>
L3	<after>, <also>, <at_least>, <before>, <beginning_of>, <both>, <but_not>, <by>, <comparative>, <confirm>, <day_num>, <day_ord>, <day_spec>, <during>, <end_of>, <end_point>, <enough_time>, <entire>, <every>, <except_for>, <floor_number>, <holiday>, <i_could_meet>, <i_have_x_free>, <index>, <last_temp>, <longer_than>, <meet>, <most_of>, <most>, <next_temp>, <part_of>, <portion>, <quantity>, <relative_point>, <rest_of>, <soon>, <**start_point**>, <time_of_day>, <that_temp>, <thats_all>, <this_temp>, <time_mod>, <time_slot>, <**time**>, <week_after_next>, <week_of>, <x_is_bad>, <you_come>, <you_could_meet>
L4	<half>, <**hour**>, <minute>, <past>, <then>, <till>, <year>

resentation is well suited to capture the natural nestedness of human language. Figure 5(b) illustrates how the highest-level <temporal> concept is propagated through *sometime between nine and five*.

The PHOENIX system does not perform a detailed, syntactic analysis of the input utterance. Expressions that are not relevant to the task at hand are simply ignored by the parser. For example, given the utterance:

> *I am busy all afternoon that Thursday* ***so if you move all the way to*** *the fourth of August*
> *I am free in the afternoon there or the morning of the fifth*

the words in **boldface** are ignored. In order to convert the semantic tree to the encoding required by the stochastic method, unlabeled words are mapped into <GARBAGE> labels, which are automatically inserted into the semantic sequence. In the example in Figure 6, the



(a)

probably	↦	<interj>
sometime	↦	<temporal><point><time_unit>
between	↦	<temporal><interval>
nine	↦	<temporal><interval><start_point><time><hour>
and	↦	<temporal><interval>
five	↦	<temporal><interval><end_point><time><hour>
would	↦	<agree>
be	↦	<agree>
good	↦	<agree>

(b)

Figure 5. Conversion of semantic trees into tree-labels to be used by the stochastic component, as exemplified for the SDU *probably sometime between nine and five would be good*; (a) PHOENIX tree representation; (b) corresponding tree-labels each representing the complete path from the root down to the leaf token in the tree.

relevant parts of the SDU *but, only, time*, etc. are matched to the tree-labels from the PHOENIX parses, whereas the irrelevant words *if, that, is* and *the* correspond to <GARBAGE> labels.

4.2. Utterance pre-processing

Stochastic methods require substantial amounts of data for the estimation of their parameters. Corpora of spoken language are still limited in size, a fact that is problematic because events rarely observed in the training data are not adequately modeled. As a result, the estimates may become unreliable. Therefore, the data sparseness requires matching the model size to the amount of training data available. In addition to back-off techniques (Katz, 1987), a category-based unification is used to reduce the input variability. Typical word categories in this domain deal with times and localities, e.g. /DAYTIME/, /LOCALITY/, /WEEKDAY/, etc. The eight unification categories employed, along with some example words, are shown in Table II. Still, compared to the information retrieval applications ATIS and MASK, this type of pre-processing is less significant in terms of parameter reduction.

Words that systematically correspond to the semantic <GARBAGE> label as they are judged to be irrelevant with respect to the ESST are called {filler} words. In the pre-processing, they are removed from the training and test data, since they do not contain nor propagate any

SDU:

but if that is the only time we can get together that will be great

Tree-label representation:

but	↦	<conj>
if	↦	<**GARBAGE**>
that	↦	<**GARBAGE**>
is	↦	<**GARBAGE**>
the	↦	<**GARBAGE**>
only	↦	<sugg_meet><temporal><point><only>
time	↦	<sugg_meet><temporal><point><time_unit>
we	↦	<sugg_meet>
can	↦	<sugg_meet>
get	↦	<sugg_meet>
together	↦	<sugg_meet>
that	↦	<agree>
will	↦	<agree>
be	↦	<agree>
great	↦	<agree>

Figure 6. <GARBAGE>-label insertion in the semantic ESST corpus for words which are judged by the PHOENIX parser to be irrelevant for the specific application.

TABLE II. Category unification in ESST

Categories	Example words
/DAYTIME/	*afternoon, evening, morning*
/LOCALITY/	*Bahamas, cafeteria, town*
/MEAL/	*breakfast, brunch, dinner, lunch*
/MONTH/	*April, August, December, February*
/NAME/	*Andrea, Andrew, Kathy, Linda*
/NUMBER/	*eight, eleven, fifteen, five*
/ORDINAL/	*eighteenth, eighth, eleventh*
/WEEKDAY/	*Monday, Saturday, Thursday*

meaningful information. However, words that correspond to <GARBAGE> in context of the specific SDU are not removed. For example,

> *and yeah* ***hopefully*** *that will work*

pre-processed into

> and yeah {**filler**} that will work

corresponds to the semantic sequence

> <conj> <agree> <**GARBAGE**> <GARBAGE> <GARBAGE> <GARBAGE>

The SDU is then transformed into

> and yeah that will work

In this example, *hopefully that will work* is considered to be irrelevant and not translated. Therefore, it corresponds to <GARBAGE>. The sequence *that will work* may be significant in a different context. Consequently, it is not pre-processed into {filler} labels.

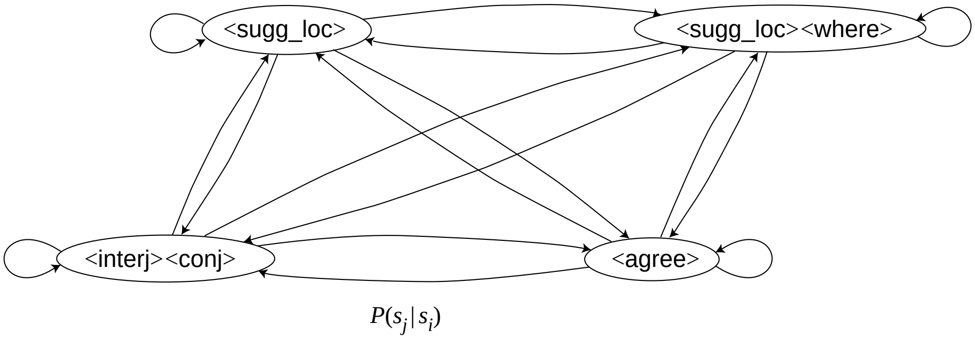


Figure 7. Ergodic semantic Markov model, all states, such as the examples <interj><conj>, <agree>, <sugg_loc> and <sugg_loc><where> shown are fully connected.

5. Stochastic model

Relative occurrences of model states and observations are used to establish the Markov model, whose topology needs to be fixed prior to training and decoding. This topology is illustrated in Figure 7. As for ATIS and MASK, semantic labels are defined as the states s_j. All states, such as the examples <interj><conj>, <agree>, <sugg_loc> and <sugg_loc><where> shown can follow each other; thus the model is *ergodic*.

Semantic decoding consists of maximizing the conditional probability $P(S|O)$ of some state sequence S given the observation sequence O. The pre-processed words in the utterance are defined as the observations o_m. Using Bayes rule, the conditional probability is re-formulated as follows:

$$[S]_{opt} = \arg\max_{\mathrm{S}}\{P(S)P(O|S)\}\cdot \tag{1}$$

Given the dimensionality of the sequence O, the direct computation of the likelihood $P(O|S)$ is intractable. However, simple recursive procedures allow us to solve this problem. They imply the estimation of HMM parameters, the bigram state transitions probabilities $A = P(s_j|s_i)$ and the observation symbol probability distribution $B = P(o_m|s_j)$ in state j at time t.

Figure 8 shows a particular path through the Markov model using the example states in Figure 5. The progression through the state sequence of semantic labels generates sequences of observations each of which represents a word in the utterance *okay that will be fine with me where would you like to meet*. Temporal progression and sequence generation are guided by the state transition and observation probabilities. They have been previously learned from a large number of correspondences between words and semantic labels in the training data. Table III shows some example state-observation correspondences. Words may be assigned to different semantic labels, e.g. *then* is associated with both, <conj> and <interj>.

Based on the model, the most likely state sequence is determined using the *Viterbi algorithm* (Rabiner & Juang, 1986). Given a significant amount of model parameters, a back-off technique (Katz, 1987) allows us to adequately estimate probabilities of rare observation and state occurrences in the training corpus.

6. Characteristics of the training corpus

The stochastic model of the understanding component has been trained using 9525 utterance transcriptions along with their sequences of semantic tree-labels. Using the JANUS prototype,

TABLE III. Examples of semantic labels in the ESST corpus along with a selection of the corresponding lexical entries

Semantic label	Example words
<agree>	*alright, good, great, perfect*
<conj>	*also, and, but, then*
<interj>	*actually, maybe, perhaps, then*
<q_your_availability><unit>	*days, hours, month, week*
<your_tum>	*maybe, what, would*
<GARBAGE>	*again, are, for, have, in*

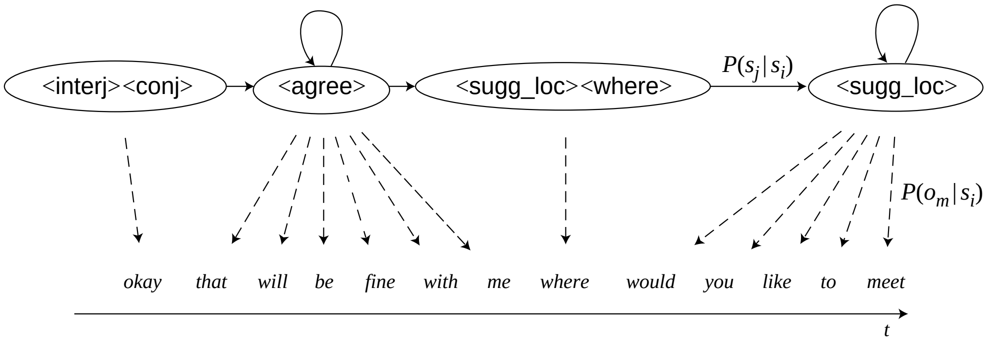


Figure 8. Semantic decoding is progressing on a path through the Markov model. It generates word sequences, the ESST example utterance *okay that will be fine with me where would you like to meet*.

the ESST data were collected at CMU, the University of Pittsburgh and Multicom (United States), the University of Karlsruhe, ETRI (Korea), UEC and ATR (Japan) (Waibel *et al.*, 1996). With 10 405 utterances used for MASK and 10 718 for ATIS the amount of the training data employed is roughly the same. [All data for MASK were collected by a real prototype system (**?**), as opposed to ATIS, where a Wizard-of-Oz set-up was used (MADCOW, 1992).]

Table IV shows characteristics of the ESST training corpus. The human-to-human dialogs result in a relatively large average utterance length (over 26 words), as well as a large lexicon size (2623 different words). In the PHOENIX system, each SDU is analysed independently. In the corpus labeled by the rule-based component, the SDU boundaries have been determined prior to training and testing (Lavie *et al.*, 1996). After this segmentation, the average length of the analysis sequences (9·3 words) is comparable to those of the information retrieval applications (8·0 for MASK and 9·1 for ATIS).

Utterance pre-processing reduces the lexicon size considerably, notably because of the relatively large number of {filler} words (1974 compared to 883 for MASK and 487 for ATIS) which are removed prior to training and decoding. This implies that PHOENIX ignores the conversational character of the negotiation dialogs. Limited to the essential parts of the utterance leads to rather terse but acceptable translations (Waibel *et al.*, 1996). The 133 basic tokens (Table I) combine to create 2711 tree-labels used as the model states (compared to 74 states for MASK and 112 states for ATIS).

TABLE IV. Characteristics of the ESST training corpus used for statistical modeling in natural language understanding

#utterances		9525
	avg. utterance length	26·5
#SDU		30 628
	avg. #SDU/utterance	3·2
	avg. #words/SDU	9·3
lexicon size		2632
	after pre-processing (model observations)	552
	#{filler} words	1974
semantic representation	*#tokens*	133
	#tree-labels (model states)	2711

7. Performance assessment

The stochastic component has been evaluated and compared in performance with the rule-based PHOENIX parser which is integrated in the JANUS speech-to-speech translation system. The test corpus consists of 258 utterance transcriptions containing 759 SDUs. The semantic accuracy was evaluated at the SDU level first, using an exact-match paradigm that compares the hypothesis and the reference on a label-by-label basis. Since only the sequential alignment of relevant semantic tree-labels is used for translation, the semantic evaluation does not account for <GARBAGE> labels. In this study, the semantic reference representation is the output of the PHOENIX parser. A human expert then analyses the incorrectly flagged sequences. If these are equivalent to, or more appropriate than, the reference, they are re-scored as correct.

For the evaluation of the translation accuracy from English to German, the SDUs are further broken down into the smaller speech-acts. Examples for speech-acts in Figure 5(a) are *probably*, *sometimes between nine and five and would be good*. In this way, more weight is given to longer SDUs, and SDUs containing both in- and out-of-domain speech-acts can be judged more accurately. Each speech-act translation is then assigned a grade by human graders as described in (Gates *et al.*, 1996). A set of consistent criteria are employed for judging the quality of the translated utterances as well as their relevance to the current domain. Assisted by grading tools, the translation is scored by one or more independent human experts. In order to obtain reliable evaluation results, several independent graders who are not involved in system development are employed to score the translations. The individual scores are averaged together to obtain the final result.

A speech-act contains semantically coherent pieces of information. Each speech-act fits as either relevant to ESST (in-domain) or not relevant (out-of-domain). In

> *okay, that's fine, so Wednesday the third at the coffee shop*

an example of an in-domain speech-act would be *so Wednesday the third at the coffee shop*; in

> *all-right, sounds like a deal, but I got hepatitis from the food the last time I ate at that coffee shop, so why don't we meet at Yum Wok at twelve. We can grab a bite to eat and then walk back to the office and go over this material for the month of May*

the phrase *I got hepatitis from the food the last time I ate at that coffee shop* is judged to be out-of-domain.

TABLE V. Semantic error as well as translation errors for training and testing the stochastic component for natural language understanding in ESST. The semantic error is given on the SDU level, the translation errors on the speech-act level

Semantic error (%)	Translation error (%)	
STOCHASTIC	STOCHASTIC	PHOENIX
18·4	30·1	23·8

In-domain phrases are labeled with one of the following grades:

- *perfect* (*p*): the system provides a fluent translation with all information conveyed
- *good* (*k*): all important information is translated correctly but some unimportant details are missing, or the translation is awkward
- *bad* (*b*): the translation is unacceptable

The global judgement of an *acceptable* translation includes *perfect* and *good* assignments.

If the speech-acts are out-of-domain, the graders can assign one of the following grades:

- *excellent* (*e*): the system provides a good translation, even though it is an out-of-domain speech-act
- *good* (*g*): the translation is non-disruptive
- *empty* (*t*): the system provides no translation at all
- *bad* (*d*): the system provides a disruptive, spurious translation of the phrase

For out-of-domain utterances, the category *acceptable* includes *excellent*, *good* and *empty* translations. The overall *translation accuracy* is calculated as the sum of acceptable in- and out-of-domain speech-act translations over the total number of speech-act translations.

7.1. Quantitative results

The results of the performance evaluation for the stochastic component for natural language understanding in ESST are given in Table V and are compared with the rule-based PHOENIX parser.

The stochastic component obtains an 18·4% error rate on the semantic representations. Both the semantic output of the stochastic component and the rule-based PHOENIX output are run through the generation module to produce German translations, the accuracy of which is then measured by the expert graders at the speech-act level. In this evaluation, the stochastic component obtains a 30·1% translation error compared to 23·8% for the rule-based parser. The fact that we used a corpus of uncorrected semantic representations produced by PHOENIX means that the stochastic implementation is limited by the inevitable shortcomings of the rule-based method. The error scores of the stochastic component are therefore relative to the performance of the PHOENIX system. In fact, the experiences in MASK (Minker, 1997) lead to the conclusion that the stochastic component is able to outperform the rule-based parser, if the training corpus is designed entirely for the stochastic method. The translation results are also likely to be influenced by the performance of the PHOENIX generator (Fig. 3), which is only optimally adapted to the rule-based method.

Table VI shows a breakdown of the translation evaluation results for both in- and out-of-domain speech-acts. Globally, for in-domain speech-acts, the reference and hypothesis translations are rarely excellent (*p*), but frequently scored correct (*k*). Interesting is the result

TABLE VI. Breakdown of the translation evaluation results (%) using independent expert graders (Gates *et al.* 1996). Grades for in-domain speech-acts are: p = perfect, k = good, b = bad, for out-of-domain speech acts: e = excellent, g = good, t = good — not translated, d = bad

	In-domain speech-acts (90·7%)				Out-of-domain speech-acts (10·3%)					Global
	p	*k*	*b*	accepted	*e*	*g*	*t*	*d*	accepted	accuracy
STOCHASTIC	2·7	69·1	28·2	71·9	11·9	4·2	43·4	40·5	59·5	**69· 9**
PHOENIX	4·2	73·8	22·0	78·0	11·2	4·2	53·0	31·6	68·4	**76·2**

for out-of-domain speech-acts: the stochastic component slightly outperforms PHOENIX in excellently scored speech-acts (e). As described in the qualitative analysis below, the system propagates significant tree-labels instead of <GARBAGE>. This reduces the number of empty (t) out-of-domain translations but implies a higher risk of an incorrect labeling of this type of speech-act. Globally, the performance gain of the stochastic component is therefore outweighed for out-of-domain speech-act translations.

7.2. Qualitative analysis

The semantic hypotheses output by the stochastic component were analysed in order to identify the strengths and weaknesses of the stochastic method when applied to the scheduling task (Fig. 9).

In the first example, the phrases *are not good, that is not good*, etc., are matched with the <give_info> <my_unavailability> label. As PHOENIX attempts to match entire expressions, the rule-based parser fails if insertions occur, e.g. *that is not any good, that is really not so good, isn't good*, etc. (*P1*). The stochastic decoding is robust. In *S1*, it identifies the isolated word *not* as a triggering reference word for <give_info><my_unavailability>, which is then successfully propagated, since the transition probabilities between labels including <give_info><my_unavailability> are high. They outweigh the weak probabilities of some unknown or less frequent insertions.

The stochastic method is flexible. Instead of triggering the <GARBAGE> labels as does PHOENIX for *we can go into* in *P2*, it propagates significant tree-labels resulting in a smooth semantic representation (in the test data the stochastic method triggers 1186 <GARBAGE> labels compared to 1659 for the rule-based method). In *S3*, the stochastic component labels the entire speech-act with the <q_availability> concept. PHOENIX proposes an alternative representation (*P3*). It labels *what time is best for you* with the <q_availability> and *on the tenth* with the <temporal> concept. The solution proposed by the stochastic method seems to be more appropriate for this segmentation.

The flexibility of the stochastic method, illustrated for *S2* and *S3*, appears to have turned out to be a drawback. In *S4*, the phrase *I will be* triggers <give_info><my_unavailability>, learned from *I will be out of town, I will be away*, etc., in the training. The phrase *the smog of* triggers <temporal><point><rest_of>, learned from *the rest of that day, the rest of this month* etc., regardless of the weak observation probabilities of *smog* and *Los*. Propagating the incorrect labels results in an erroneous annotation of the entire speech-act.

Figure 10 shows examples of graded English-to-German speech-act translations produced on the basis of the semantic representations, output of the stochastic component:

- The translation of speech-act *T1* is graded excellent in-domain (p).
- The translation of *T2* contains a false start (leider kann ich leider bin ich) and some ungrammatical phrases (und am zweite Juni nicht). But since it contains the information conveyed, it is still scored correct in-domain (k).

S1:	*that is not any good*	↦	<give_info><my_unavailability><that_wont_work>
P1:	*that is not any*	↦	<GARBAGE>
	good	↦	<agree>
S2:	*we can go*	↦	<sugg_meet>
	into the	↦	<sugg_meet><temporal><point>
	evenings	↦	<sugg_meet><temporal><point><time_of_day>
	or	↦	<conj>
	the	↦	<sugg_meet><temporal><point><next_temp>
	weekends	↦	<sugg_meet><temporal><point><day_of_week>
P2:	*we can go into*	↦	<GARBAGE>
	the	↦	<sugg_meet><temporal><point>
	evenings	↦	<sugg_meet><temporal><point><time_of_day>
	or	↦	<conj>
	the	↦	<sugg_meet><temporal><point>
	weekends	↦	<sugg_meet><temporal><point><day_of_week>
S3:	*what time*	↦	<q_availability><when>
	is best for you	↦	<q_availability>
	on	↦	<q_availability><point>
	the	↦	<q_availability><point><date>
	tenth	↦	<q_availability><point><date><day_ord>
P3:	*what time*	↦	<q_availability><when>
	is best for you	↦	<q_availability>
	on	↦	<temporal><point>
	the	↦	<temporal><point><date>
	tenth	↦	<temporal><point><date><day_ord>
S4:	*I will be choking on*	↦	<give_info><my_unavailability>
	the smog of Los	↦	<give_info><my_unavailability><temporal><point><rest_of>
	Angeles	↦	<give_info><my_unavailability><temporal><point><time_unit>
P4:	*I will be chocking on*	↦	<GARBAGE>
	the smog of Los	↦	<GARBAGE>
	Angeles	↦	<GARBAGE>

Figure 9. Parsing examples demonstrating the strengths and the weaknesses of the stochastic method in ESST.

- The translation of speech-act *T3* is scored erroneous in-domain (*b*).
- Politeness forms are considered as out-of-domain. *G4* is therefore an excellent (*e*) translation.
- The translation of speech-act *T5* is graded good out-of-domain (*g*).
- The phrase *later alligator* (*T6*) is out-of-domain. Assigned to the <GARBAGE> state, it is not translated and therefore scored good (*t*).
- *G7* is an example for an incorrect translation of an out-of-domain speech-act. As discussed earlier, the stochastic component assigns more significant semantic labels than the rule-based decoder, implying the higher risk of incorrect labeling and translation.

T1:	*Thursday the nineteenth sounds good*	
G1:	Ja Donnerstag den neunzehnten geht es bei mir ganz gut	*p*
T2:	*I am out of town from May thirty first until the second of June*	
G2:	Leider kann ich leider bin ich außer Haus zwischen Mai ein und dreißigsten und am zweite Juni nicht (lit. *unfortunately I can unfortunately I am out of town between May thirty first and on second of June not*)	*k*
T3:	*is that the eleventh*	
G3:	also gut machen es wir den elften alles klar (lit. *okay let's do it the eleventh all-right*)	*b*
T4:	*bye*	
G4:	auf wiedersehen	*e*
T5:	*no why don't you write it down for me*	
G5:	nein (lit. *no*)	*g*
T6:	*later alligator*	
G6:	–	*t*
T7:	*by the way your secretary says you got a headache; you know what you got to do is just get away from everything*	
G7:	was meinen Sie dazu (lit. *what do you think about it*)	*d*

Figure 10. Examples of graded speech-act translations. The significance of the grading assistant grades are for in-domain speech-acts: p = perfect, k = good, b = bad, for out-of-domain speech-acts: e = excellent, g = good, t = good–not translated, d = bad. If necessary, the corresponding literal translation is given in parenthesis.

8. Summary and Conclusions

In this article, we have reported experiences in porting a stochastic component for natural language understanding from a setting of human–machine interaction to that of a human–human interaction. Tests were performed in the context of an appointment scheduling task, the ESST.

The use of the stochastic component for natural language understanding (repetition) involved porting the method to a substantially different domain. Compared with the simpler semantic frames used in ATIS and MASK, the PHOENIX representation of a tree-based case grammar has been adapted to the stochastic method. The derivation of tree-labels to model the nestedness of human language supports a more efficient propagation of semantic information. The study shows that domain and language porting of such a method is relatively straightforward and that it is sufficient to train the system on data sets based on a semantic formalism which is appropriate for the application and language.

Comparative performance tests were carried out using the stochastic component and the rule-based PHOENIX parser. As in the ATIS task, the performance of the stochastic component in ESST is likely to have been limited through the use of a semantic corpus which is not a product of the component itself but is generated using a rule-based system. Even though this suboptimal semantic corpus was used, the stochastic decoder obtains reasonable semantic and translation errors (18·4% and 30·1%, respectively). Qualitatively, the stochastic method allows for a robust decoding through modeling of isolated words, as opposed to the rule-based parser in which particular, task-dependent expressions need to be defined by hand. These represent an over-specialization, since the system fails if insertions occur within these expressions. The stochastic method is also flexible: it creates smooth semantic representations through labeling and propagating a maximum amount of significant labels. However, this risk-taking strategy is penalized by an increase in incorrectly translated out-of-domain speech-acts. The qualitative evaluations have demonstrated that, similar to ATIS and MASK, the stochastic decoding is robust and flexible. However it is not risk-avoiding. It propagates meaningful tree-labels instead of <GARBAGE> labels. This reduces the number of empty out-of-domain translations, but implies a higher risk of incorrect concept triggering and error propagation.

Further improvements may be achieved if the training corpus was entirely designed for the stochastic method and more training data were available given the significant number of model observations. As concluded from the experiences in MASK, the design of the stochastic component focuses on the creation of a semantic corpus using an iterative labeling approach. By adapting the semantic labels to the method, the stochastic component outperforms the rule-based parser. Also, to be able to evaluate the translation accuracy, the language generation component should be redesigned to be optimally adjusted to the encoding used by the stochastic method.

The authors gratefully acknowledge the contributions of Donna Gates, Sham Gardner and Peter Zerfass, who provided assistance in the experiments and evaluations reported in this article.

References

Bruce, B. (1975). Case systems for natural language. *Artificial Intelligence*, **6**, 327–360.

Bub, W. T. & Schwinn, J. (1996). Verbmobil: the evolution of a complex large speech-to-speech translation system. *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, PA, pp. 2371–2374.

Finke, M., Geutner, P., Hild, H., Kemp, T., Ries, K. & Westphal, M. (1997). The Karlsruhe-Verbmobil speech recognition engine. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, pp. 83–86.

Gates, D., Lavie, A., Levin, L., Waibel, A., Gavaldà, M., Mayfield, L., Woszczyna, M. & Zahn, P. (1996). End-to-end evaluation in JANUS: a speech-to-speech translation system. *Proceedings of ECAI*, Budapest, Hungary, pp. 35–40.

MADCOW (1992). Multi-site data collection for a spoken language corpus. *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, pp. 7–14.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**, 400–401.

Lavie, A., Gates, D., Coccaro, N. & Levin, L. (1996). Input segmentation of spontaneous speech in JANUS: a speech-to-speech translation system. *Proceedings of the ECAI*, pp. 54–59 .

Lavie, A., Waibel, A., Levin, L., Finke, M., Gates, D., Gavaldà, M., Zeppenfeld, T. & Zhan, P. (1997). JANUS III: speech-to-speech translation in multiple languages. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, pp. 99–102.

Levin, E. & Pieraccini, R. (1995). CHRONUS — the next generation. *Proceedings of the ARPA/Workshop on Spoken Language Systems Technology*, Austin, TX, pp. 269–271.

Mayfield, L., Gavaldà, M., Ward, W. & Waibel, A. (1995). Concept-based speech translation. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Detroit, MI, pp. 97–100.

Minker, W. (1997). Stochastically-based natural language understanding across tasks and languages. *Proceedings of Eurospeech*, Rhodes, Greece, pp. 1423–1426.

Minker, W., Bennacef, S. K. & Gauvain, J. L. (1996). A stochastic case frame approach for natural language understanding. *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, PA, pp. 1013–1016.

Rabiner, L. R. & Juang, B. H. (1986). An introduction to hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **3**, pp. 4–16.

Schwartz, R., Miller, S., Stallard, D. & Makhoul, J. (1996). Language understanding using hidden understanding models. *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, PA, pp. 997–1000.

Waibel, A. (1996). Interactive translation of conversational speech. *Computer*, **29**, 41–48.

Waibel, A. *et al*. (1996). JANUS-II-translation of spontaneous conversational speech. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Atlanta, GA, pp. 409–412.

Ward, W. (1994). Extracting information in spontaneous speech. *Proceedings of the International Conference on Spoken Language Processing*, Yokohama, Japan, pp. 83–86.

Ward, W. & Issar, S. (1995). The CMU ATIS system. *Proceedings of the ARPA Workshp on Spoken Language Systems Technology Workshop*, Austin, TX, pp. 249–251.

(Received 12 November 1997 and accepted for publication 27 January 1999)