Hidden Understanding Models for Machine Translation

Wolfgang Minker[1], *Marsal Gavaldà*[2], *Alex Waibel*[3]

[1]Spoken Language Processing Group, LIMSI-CNRS (France), minker@limsi.fr
[2]Language Technologies Institute, Carnegie Mellon University (USA), marsal@cs.cmu.edu
[3]Carnegie Mellon University (USA) and University of Karlsruhe (Germany), ahw@cs.cmu.edu

ABSTRACT

We demonstrate the portability of a stochastic method for understanding natural language from a setting of human-machine interactions (ATIS - Air Travel Information Services and MASK - Multimodal Multimedia Automated Service Kiosk) into the more open one of human-to-human interactions. The application we use is the English Spontaneous Speech Task (ESST) for multilingual appointment scheduling. Spoken language systems developed for this task translate spontaneous conversational speech among different languages.

1. INTRODUCTION

In this paper, a stochastic component for natural language understanding, initially developed as a part of a spoken language system for the information retrieval applications ATIS (Air Travel Information Services) and MASK (Multimodal Multimedia Automated Service Kiosk) [7], is ported to a multilingual, appointment scheduling task, the English Spontaneous Speech Task (ESST) [8].

Machine translation systems combine speech recognition, natural language understanding and dialog to capture 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 this study, statistical modeling techniques replace the commonly-used *manually* generated grammar rules that parse the recognizer output into a semantic representation. The statistical models are derived from the *automatic* analyses of large corpora of naturally-occurring sentences along with their semantic representations. Such stochastic methods have been applied in the BBN-HUM [10] and the AT&T-CHRONUS [5] systems for the American ARPA-ATIS task. To date, the language and domain portability of stochastic parsers has not been investigated. However, portability and flexibility issues, discussed in this paper, represent the essential motivations for applying a stochastic method for the semantic analysis.

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 [11], transforms the output of the recognizer into semantic trees. 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.

In the next sections, we describe how both the rule-based (PHOENIX) and the stochastic parsers work (Sections 2 and 3) and introduce the semantic representation (Section 4). In Section 5 the stochastic model 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 translation modules used in JANUS [6, 12] 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 in which the left-hand sides (rule 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 1 shows some sample grammar rules for a scheduling domain. Note that grammar terminals (i.e., lexical items) and grammar nonterminals (i.e., semantic tokens) are freely mixed in the right-hand sides. Auxiliary nonterminals (uppercased in the example) are used only as a short-hand for the grammar writer and do not appear in the final parse tree.

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

Figure 1: Sample grammar rules for ESST [8]. Lexical items (in lower-case) and calls to semantic nets (in uppercase or enclosed in square parentheses) are freely mixed. A '*' indicates optional token, a '+' indicates repeatable 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, good bye, bye bye, etc.*

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 nonterminals 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.

Once the grammar is deemed developed enough, it is complied by PHOENIX into Recursive Transition Networks (RTN), each grammar nonterminal giving rise to one RTN. A subset of the nonterminals 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 preprocesses it by eliminating out-of-vocabulary words (i.e., words not appearing in the grammar) and expanding some contractions (e.g. *I'm* $\rightarrow$ *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. STOCHASTIC COMPONENT

The semantic analyzer using a stochastic method is based on Hidden Markov Modeling (HMM). The functional diagram is given in Figure 2: 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 [9], outputs the most likely semantic representation given a transcribed utterance as input.

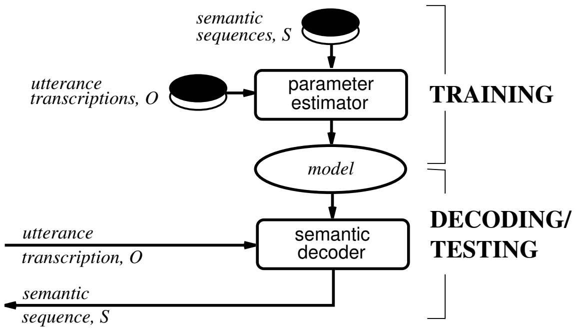


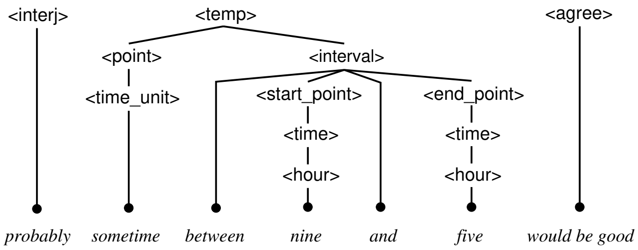
Figure 2: Overview of the stochastic parsing component [8].

The stochastic component uses the same techniques for training and decoding that were developed for ATIS and MASK [7], 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.

4. KNOWLEDGE REPRESENTATION

The parameters of the stochastic model are estimated given preprocessed word sequences along with their semantic representation.

Semantic representation The trees generated by PHOENIX are illustrated in Figure 3(a). The parser tries to model the relevant ESST information structures as well as the lexical realization of these structures in various languages. A set of *semantic tokens* corresponds to concepts and sub-concepts in an utterance. These semantic tokens can be seen as the vertices of a directed acyclic graph in which the edges refer to concept-subconcept relations. For example in Figure 3(a), a typical temporal concept <temp> 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. For instance in Figure 3(a), the utterance is parsed into the independent semantic speech-acts <interj>, <temp> and <agree> that capture the top-level meaning of *probably*, *sometime between nine and five* and *would be good*.



(a)

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

(b)

Figure 3: Conversion of semantic trees into tree-labels exemplified for *probably sometime between nine and five would be good* [8]; (a) PHOENIX tree-representation, (b) corresponding tree-labels to be used by the stochastic parsing component. Each label represents the complete path from the root down to the leaf token in the tree.

The rule-based PHOENIX output is not in a form that can be directly used by the model parameter estimator of the stochastic component. Each individual word of the input utterance must have a corresponding *semantic label*. The tree-based representation (Figure 3(a)) was converted into sequences of semantic *tree-labels* (Figure 3(b)). A tree-label represents the complete path from the root down to the lowest level token. An example path trough the tree is <temp> $\rightarrow$ <interval> $\rightarrow$ <start_point> $\rightarrow$ <time> $\rightarrow$ <hour> $\mapsto$ *nine*.

The PHOENIX system does not perform a detailed, syntactic analysis of the input utterance. Spontaneous speech effects, ill-formed parts-of-speech and expressions that are not relevant to the task at hand, are simply ignored by the parser. E.g., in *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 a <GARBAGE> label, and automatically inserted into the semantic sequence.

Utterance preprocessing The transcriptions of the utterances were broken down into smaller *SDU*s (Semantic Discourse Units) using a combination of acoustic, lexical, semantic and statistical knowledge sources, as described by Lavie [4].

Stochastic methods require substantial amounts of data for the estimation of their parameters. As the data sparseness requires matching the model size to the amount of training data available, a category-based unification is used in order to reduce the input variability. The eight word categories employed in this domain are /DAYTIME/, /LOCALITY/, /MEAL/, /MONTH/, /NAME/, /NUMBER/, /ORDINAL/ and /WEEKDAY/.

Words that systematically correspond to the semantic <GARBAGE> label as they are judged to be irrelevant with respect to ESST are called {filler} words. In the preprocessing, the {filler} - <GARBAGE> correspondences are removed from the training sequences, since they do not contain nor propagate any meaningful information. However, words that correspond to <GARBAGE> only in context of a specific SDU are not removed.

5. STOCHASTIC MODEL

Semantic decoding consists of maximizing the conditional probability $P(S|O)$ of some state sequence S given the observation sequence O. Using Bayes rule, this probability is reformulated as follows:

$$[S]_{opt} = \operatorname*{argmax}_{S} \{P(S)P(O|S)\} \qquad (1)$$

Given the dimensionality of the sequence O, the direct computation of the likelihood $P(O|S)$ is intractable. However, simple recursive procedures allow to solve this problem. They imply the estimation of Hidden Markov Model (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.

Figure 4 illustrates the topology of the Markov Model established by the parameter estimator for use by the semantic decoder (Figure 1). Semantic tree-labels are defined as the states s_j, the preprocessed words as the observations o_m. The example states <interj><conj>, <agree>, <sugg_loc> and <sugg_loc><where> can follow each other, the model is *ergodic*.

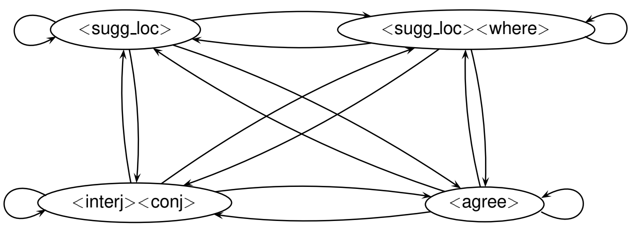


Figure 4: Ergodic semantic Markov Model; the states <interj><conj>, <agree>, <sugg_loc> and <sugg_loc><where> are fully connected [8].

Based on the stochastic model, the most likely state sequence is determined using the *Viterbi algorithm* [9]. Given a significant amount of model parameters, a back off technique [2] allows to adequately estimate probabilities of rare observation and state occurrences in the training corpus.

6. TRAINING CORPUS

The stochastic model of the understanding component has been trained using 9,525 utterance transcriptions along with their sequences of semantic tree-labels. The human-to-human dialogs result in a relatively large average utterance length (over 26 words), as well as a large lexicon size (2,623 different words). In the PHOENIX system each SDU is analyzed independently. In the corpora, the SDU boundaries have been determined prior to training and testing [4]. After this segmentation, the average length of the analysis sequences is reduced to 9.3 words.

Utterance preprocessing reduces the lexicon size considerably (552 different words), notably because of the relatively large number of {filler} words which are not considered. This implies that PHOENIX ignores the conversational character of the negotiation dialogs. Focalizing on the essential parts of the utterance leads to rather terse but acceptable translations [11]. The 133 basic semantic tokens combine to create 2,711 tree-labels.

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 analyzes 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 have been further broken down into the smaller speech-acts. The corresponding semantic sequences have been translated using the PHOENIX generation module. Each speech-act translation is then assigned a grade by human graders as described in [1].

Quantitative results The performance evaluation results of the stochastic component for natural language understanding in ESST are given in Table 1 and are compared to those obtained by the rule-based PHOENIX parser.

Semantic error (%)	*Translation error (%)*	
STOCHASTIC	STOCHASTIC	PHOENIX
18.4	30.1	23.8

Table 1: Semantic error (on the SDU level) as well as translation errors (on the speech-act level) obtained by the stochastic component for natural language understanding in ESST.

The stochastic component obtains a 18.4% semantic error rate on the SDU level. In the translation evaluation, it obtains a 30.1% speech act error compared to 23.8% measured 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 PHOENIX. In fact, the experiences in MASK [7] lead to the conclusion that the stochastic component is able to outperform the rule-based parser, if the training corpus is entirely designed for the stochastic method. The translation results are also likely to be influenced by the performance of the PHOENIX translation generator which is again optimally adapted to the rule-based parser.

Qualitative analysis The semantic hypotheses output of the stochastic parser were analyzed in order to identify the strengths and weaknesses of the method (Figure 5).

In the first example, the phrases *are not good, that is not good, etc.*, need to be matched with the <my_unavail> 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*). In turn, the stochastic decoding is robust. In *S1*, it identifies the isolated word *not* as a reference word for <my_unavail>, which is then successfully propagated, since the transition probabilities between labels including <my_unavail> are

S1:<my_unavail><wont_work>	*that is not any good*
P1:<GARBAGE>	*that is not any*
<agree>	*good*
S2:<sugg_meet>	*we can go*
<sugg_meet><temp><point>	*into the*
<sugg_meet><temp><point><time_day>	*evenings*
<conj>	*or*
<sugg_meet><temp><point><next_temp>	*the*
<sugg_meet><temp><point><day_week>	*weekends*
P2:<GARBAGE>	*we can go into*
<sugg_meet><temp><point>	*the*
<sugg_meet><temp><point><time_day>	*evenings*
<conj>	*or*
<sugg_meet><temp><point>	*the*
<sugg_meet><temp><point><day_week>	*weekends*
S3:<my_unavail>	*I will be choking on*
<my_unavail><temp><point><rest_of>	*the smog of Los*
<my_unavail><temp><point><time_unit>	*Angeles*
P3:<GARBAGE>	*I will be chocking on*
<GARBAGE>	*the smog of Los*
<GARBAGE>	*Angeles*

Figure 5: Parsing examples demonstrating the strengths and the weaknesses of the stochastic method in ESST, *S* = stochastic parser, *P* = PHOENIX parser.

high. They outweigh the weak probabilities of some unknown or less frequent insertions.

The stochastic method is also flexible. Instead of triggering <GARBAGE> labels as does PHOENIX for *we can go into* in *P2*, it propagates the significant <sugg_meet> resulting in a smooth semantic representation (in the test data the stochastic method triggers 1,186 <GARBAGE> labels compared to 1,659 for the rule-based method).

The flexibility of the stochastic method, illustrated in *S2*, appears to have turned out to be a drawback. In *S3*, the phrase *I will be* triggers <my_unavail>, learned from *I will be out of town, I will be away, etc*., in the training. The phrase *the smog of* triggers <temp><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.

8. SUMMARY AND CONCLUSIONS

In this paper 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 English Spontaneous Scheduling Task (ESST).

The use of the stochastic component for natural language understanding in ESST involved porting the method to a substantially different domain. Compared to the simpler semantic frames used in ATIS and MASK, the PHOENIX semantic tree representations have been adapted to the stochastic method. The derivation of tree-labels models the nestedness of human language and therefore supports a more efficient propagation of semantic information. The study shows that task domain and language porting of a stochastic method is relatively straightforward. 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 evaluations in this domain were carried out using the stochastic component and the rule-based PHOENIX parser. Even though a suboptimal semantic training corpus (generated by PHOENIX) was used, the stochastic decoder obtains reasonable semantic and translation errors (18.4% and 30.1% respectively). Qualitatively, the stochastic method enables 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 as it implies a higher risk of incorrect concept triggering and error propagation.

Certain aspects of the presented method could be further investigated and expanded. 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 [7], 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 translation generation component should be redesigned to be optimally adjusted to the encoding performed by the stochastic parser.

The initial use of statistical modeling for semantics was not integrated with the speech recognition component. A first step in this direction would be to compare the component performance using the corrected utterance transcriptions with those obtained when using real speech recognizer output either for component training and testing or testing only.

9. ACKNOWLEDGEMENTS

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 paper.

10. REFERENCES

[1] D. Gates, A. Lavie, L. Levin, A. Waibel, M. Gavaldà, L. Mayfield, M. Woszczyna, and P. Zahn. End-to-end Evaluation in JANUS: a Speech-to-speech Translation System. *Proc. ECAI'96*.

[2] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[3] A. Lavie and M. Tomita. GLR*- An Efficient Noise Skipping Parsing Algorithm for Context Free Grammars. *Proc. IWPT'93*.

[4] A. Lavie, D. Gates, N. Coccaro, and L. Levin. Input Segmentation of Spontaneous Speech in JANUS: a Speech-to-speech Translation System. *Proc. ECAI'96*.

[5] E. Levin and R. Pieraccini. CHRONUS - The Next Generation. *Proc. ARPA-HLT'95*.

[6] L. Mayfield, M. Gavaldà, W. Ward, and A. Waibel. Concept-based Speech Translation. *Proc. ICASSP'95*.

[7] W. Minker. Stochastic versus Rule-based Speech Understanding for Information Retrieval. *Speech Communication*, 25(4):223–247.

[8] W. Minker, M. Gavaldà, and A. Waibel. Stochastically-based Semantic Analysis for Machine Translation. *Computer Speech and Language*, 13(2):177–194.

[9] L.R. Rabiner and B.H. Juang. An introduction to Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 3(1):4–16, 1986.

[10] R. Schwartz, S. Miller, D. Stallard, and J. Makhoul. Language Understanding Using Hidden Understanding Models. *Proc. ICSLP'96*.

[11] A. Waibel. Interactive Translation of Conversational Speech. *Computer*, 27(7):41–48, 1996.

[12] W. Ward. Extracting Information in Spontaneous Speech. *Proc. ICSLP'94*.