ABSTRACT
We describe the first sentence understanding system that is completely based on learned methods both for understanding individual sentences, and determining their meaning in the context of preceding sentences. We divide the problem into three stages: semantic parsing, semantic classification, and discourse modeling. Each of these stages requires a different model. When we ran this system on the last test (December, 1994) of the ARPA Air Travel Information System (ATIS) task, we achieved 13.7% error rate. The error rate for those sentences that are context-independent (class A) was 9.7%.

1. INTRODUCTION
Language understanding systems that use a large set of rules to explain the syntactic and semantic possibilities for spoken sentences suffer from a lack of robustness when faced with the wide variety of spoken sentences that people really use.

One reason for this lack of robustness is that rule based systems must compromise between coverage and overgeneration. Coverage can be increased by writing new rules, or by relaxing the preconditions on existing rules. But each of these alternatives increases the likelihood of spuriously triggering an incorrect rule. Conversely, overgeneration can be reduced by removing rules, or by making the preconditions of existing rules more specific. But these alternatives increase the likelihood of a rule not being available when it is needed. In general, the approach taken in constructing rule-based systems is to increase coverage until the damage inflicted from additional rules outweighs the benefit of those rules. This process requires substantial expertise, and usually leaves some of the domain uncovered.

Although statistical approaches have been used to solve small parts of the problem, such as part-of-speech taggers [1] and probabilistic syntactic parsers [2], most systems are still fundamentally rule-based, using these statistical approaches only to provide the input or to decide among choices that are already within the model. Levin and Pieraccini [3] used a semantic tagging process to decide on the likely semantic uses of words within a sentence. But still, they require a significant rule base to make sense out of the sequence of semantic events. [4] has developed a system that translates word sequences for context-independent sentences into unambiguous meanings but the reported error rate is quite high (25% to 30%) and they have not included any mechanism for modeling the discourse effects of previous sentences on the meaning of the current sentence.

We have developed what we believe to be the first system for spoken language understanding that is based completely on trained statistical models, derived from annotated corpora. The annotation is relatively straightforward, since it is largely based on a simple representation of the semantics of the domain. We call the system a Hidden Understanding Model (HUM) because, as with the HMM used in speech, each state in the model has all possible outputs, and we search for the most likely meaning given the input words, according to our model. (Of course the model must be redesigned for the problems in language understanding.) This system is easily and naturally combined with the n-best speech recognition answers output from a speech recognition component.

![Diagram](image)

Figure 1: Elements of statistical modeling system.

2. METHODOLOGY
In principle, the problem of understanding a sentence is simply a matter of finding the meaning, $M_d$, of a sentence in the context of a discourse, given the sequence of words, $W$, and the discourse history, $H$. This is $P(M_d | W, H)$. We factor the problem of establishing meaning into different levels with an appropriate model for each part of the problem (see Appendix A for a complete derivation). These models capture the basic hierarchical nature of language. We use four stages in our
model: Semantic Parsing, Semantic Classification, Discourse, and Query Generation. Each of these stages, except the final Query Generation stage, is modeled as a stochastic process that associates input expressions with output expressions. For example, in semantic parsing, the input expressions are sentences and output expressions are semantic parse trees.

Figure 1 shows the elements of a statistical modeling system that associates inputs with outputs. Our overall approach contains three such systems, one for each stage in our model. Initially, each system is trained by presenting it with pairs of input and output expressions. Later, when that system is presented with a new input expression, it should be capable of producing the correct output expression.

The critical elements of each statistical modeling system are:

1. Notational systems for representing input expressions and output expressions.

2. A statistical model that is capable of representing the association between input expressions and output expressions.

3. An automatic training program that, given a set of training examples, can estimate the parameters of the statistical model.

4. An decoding program that can search the statistical model to find the most likely output expression given an input expression.

3. SEMANTIC PARSING

The purpose of the semantic parse is to model how the user said something. The first stage assumes that the word sequence, $W$, was generated by a probabilistic semantic grammar. We find the n-best semantic parses (trees), $T$, that are most likely to have generated the word sequence, according to the measure $P(T) P(W | T)$. Figure 2 shows an example of a semantic parse tree used by our system.

The model of word sequences reflects the nested structure of language. For each nonterminal in the language we have an ngram language model on the sequence of symbols that it produces. The probabilities in the ngram model depend on the particular nonterminal. This model was first proposed by Seneff [5]. We have incorporated several minor differences that provide a stronger model while being somewhat more robust to lack of training data. Thus, the prior probability, $P(T)$, is modeled by the ngram transition probabilities on the nonterminals, while $P(W | T)$ is modeled by the ngram probabilities on the words at the leaves of the tree.

We have decreased the cost of annotating a large number of sentences substantially by using a commonly used semi-automatic procedure in which we use the decoder to find the most likely parse for each sentence, and then have a human verify the annotation, which is much faster than entering the annotations manually.

The estimated probabilities are smoothed in two ways. The node labels contain both semantic and syntactic labels. We back off to a model that treats the semantic and syntactic labels independently, which requires much less training. In addition, we back off from a trigram to a bigram and unigram according to the formula in Placeway [6].

4. SEMANTIC CLASSIFICATION

The semantic trees are not, in themselves, unambiguous meanings. Given the set of candidate semantic parses, $T$, we must find the n-best surface meaning frames, $M_s$, that maximize $P(M_s, T) P(W | T)$. To represent meaning we use a simple

![Figure 2: A semantic parse tree.](image-url)
nested frame language, as shown in Figure 3. We treat this as a
semantic classification problem. We determine which type of
frame we are dealing with by rescoring the semantic parse with
probability estimates that depend on the particular type of
frame.

<table>
<thead>
<tr>
<th>Air-Transportation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show: (Arrival-Time)</td>
</tr>
<tr>
<td>Origin: (City &quot;Boston&quot;)</td>
</tr>
<tr>
<td>Destination: (City &quot;Atlanta&quot;)</td>
</tr>
</tbody>
</table>

Figure 3: A sample semantic frame.

Next, we consider what to do with each concept associated with
a preterminal in the semantic parse. Each concept can be used
to fill any of the slots in the frame, while many concepts do not
fill any slot. We use a statistical decision tree as used by [2] to
find the probability of each possible action. The decision tree
can ask questions about the semantic and syntactic categories of
symbols that are up to two siblings to the left and right, and four
immediate ancestors -- up the tree.

5. DISCOURSE

The meaning can be changed due to preceding history, \( H \).
Based on the relations between current and previous frames, we
determine the most probable after-discourse frame, maximizing
\( P(Md | H, Ms) P(Ms, T) P(W | T) \). We assume that each
meaning frame has two stages: the surface meaning, \( Ms \), derived
directly from the sentence (what you said), and the discourse
meaning, \( Md \), in context (what you meant).

We annotated about 4,500 sentences with the before-discourse
and after-discourse frames. For the present, our discourse
model is concentrated on deciding which of the fields should be
inherited from the previous related sentence. We compute the
probability that we would inherit a particular field, given that it
is previously specified and is not mentioned in the current
frame, and given each of the other fields that have been
modified or not modified. Using Bayes' rule and an
independence assumption, we get

\[
P(\text{inhibit}_i | \text{MOD}) = P(\text{inhibit}_i) \prod_j P(\text{mod}_j | \text{inhibit}_i)
\]

where \( \text{mod}_j \) is the binary value of whether field \( j \) has been
modified (initialized or changed), and \( \text{inhibit}_i \) is the binary
value of whether field \( i \) should be inherited (or reset). We find
that, on held out data, only 5.0% of all the after-discourse
frames have one or more incorrect fields, when starting with the
correct before discourse frame.

6. GENERATING DATABASE QUERIES

Given the final, unambiguous semantic frame, we produce an
SQL expression that carries out the request implied by the
meaning. While this process should not introduce any errors,
we found that the rules for creating answers (the "Principles of
Interpretation") for ATIS are quite complicated. After spending
one month on writing this translation, we find that 3.5% of the
meanings are not translated correctly. We expect that this
number can be reduced with additional work.

7. EXPERIMENTS

For our experiments we used the ARPA Air Travel Information
System (ATIS) domain, which consists of sentences collected
from naive users [7] [8]. The complete HUM system was
trained on 4,500 annotated ATIS2 and ATIS3 utterances and
was run on whole sessions -from words to answers - on the
December 1994 test set of the ARPA ATIS (Air Travel
Information Service) task. The error rate was 13.7%, of which
3.5% were wrong because the translation to SQL failed. On the
"class A" subset of the sentences, which were annotated as
context-independent, the error rate was 9.7%.

8. SUMMARY

We have developed a sentence understanding system that is
completely based on learned statistical models. The system uses
three statistical models for semantic parsing, semantic
classification, and discourse processing with a different
statistical model that is appropriate for each level. While we do
not claim that the models used are necessarily the best ones, the
results are quite satisfying, in that a method based completely
on statistical methods resulted in reasonable performance.

APPENDIX A: MODEL DERIVATION

Given a string of input words \( W \) and a discourse history \( H \),
the task of a statistical language understanding system is to search
among the many possible discourse-dependent meanings \( Md \)
for the most likely meaning \( M_d \):

\[
M_d = \arg \max_{M_d} P(M_d | W, H).
\]

Directly modeling \( P(M_d | W, H) \) is difficult because the gap that
the model must span is large. A common approach in non-
statistical natural language systems is to bridge this gap by
introducing intermediate representations such as parse structure
and pre-discourse sentence meaning. Introducing these
intermediate levels into the statistical framework gives:

\[
M_0 = \arg \max_{M_0} \sum_{M_d, T} P(M_d | W, H, M_s, T) P(M_s, T | W, H)
\]

where \( T \) denotes a semantic parse tree, and \( M_s \) denotes pre-
discourse sentence meaning. This expression can be simplified
by introducing two independence assumptions:

1. Neither the parse tree \( T \), nor the pre-discourse meaning \( M_s \),
depends on the discourse history \( H \).
2. The post-discourse meaning \( M_d \) does not depend on the
words \( W \) or the parse structure \( T \), once the pre-discourse
meaning \( M_s \) is determined.

Under these assumptions,
\[ M_0 = \arg\max_{M_0} \sum_{M_S, T} P(M_D | H, M_S) P(M_S, T | W) . \]

Next, the probability \( P(M_S, T | W) \) can be rewritten using Bayes rule as:

\[ P(M_S, T | W) = \frac{P(M_S, T) P(W | M_S, T)}{P(W)} , \]

leading to:

\[ M_0 = \arg\max_{M_0} \sum_{M_S, T} P(M_D | H, M_S) \frac{P(M_S, T) P(W | M_S, T)}{P(W)} . \]

Now, since \( P(W) \) is constant for any given word string, the problem of finding meaning \( M_D \) that maximizes

\[ \sum_{M_S, T} P(M_D | H, M_S) \frac{P(M_S, T) P(W | M_S, T)}{P(W)} \]

is equivalent to finding \( M_D \) that maximizes

\[ \sum_{M_S, T} P(M_D | H, M_S) P(M_S, T) P(W | M_S, T) . \]

Thus,

\[ M_0 = \arg\max_{M_0} \sum_{M_S, T} P(M_D | H, M_S) P(M_S, T) P(W | M_S, T) . \]

We now introduce a third independence assumption:

3. The probability of words \( W \) does not depend on meaning \( M_S \) given that parse \( T \) is known.

This assumption is justified because the word tags in our parse representation specify both semantic and syntactic class information. Under this assumption:

\[ M_0 = \arg\max_{M_0} \sum_{M_S, T} P(M_D | H, M_S) P(M_S, T) P(W | T) \]

Finally, we assume that most of the probability mass for each discourse-dependent meaning is focused on a single parse tree and on a single pre-discourse meaning. Under this (Viterbi) assumption, the summation operator can be replaced by the maximization operator, yielding:

\[ M_0 = \arg\max_{M_0} \max_{M_S, T} \left( P(M_D | H, M_S) P(M_S, T) P(W | T) \right) \]

This expression corresponds to the computation actually performed by our system.

**ACKNOWLEDGEMENTS**

This work was supported by the Advanced Research Projects Agency and monitored by the Office of Naval Research under Contract No. N00014-91-C-0115, and by Pt. Huachuca under Contract Nos. DABT63-94-C-0061 and DABT63-94-C-0063. The content of this information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

**REFERENCES**


