Language Understanding Using Hidden Understanding Models

Richard Schwartz, Scott Miller, David Stallard, John Makhoul

BBN Systems and Technologies
70 Fawcett Street
Cambridge MA 02138
Schwartz@bbn.com

ABSTRACT

We describe the first sentence understanding system that is completely based on learned methods both for understanding individual sentences, and determinig their meaning in the context of preceding sentences. We describe the models used for each of three stages in the understanding: semantic parsing, semantic classification, and discourse modeling. When we ran this system on the last test (December, 1994) of the ARPA Air Travel Information System (ATIS) task, we achieved 14.5% error rate. The error rate for those sentences that are context-independent (class A) was 9.5%.

1. Introduction

The problem of understanding spoken language is one that has challenged us for many years. 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 is that, for most limited domains, a traditional syntactic explanation of a sentence is often much more complicated than the direct explanation of the meaning of the sentence in terms of the words spoken and the relations between the words.

Recently, statistical approaches have found their way into natural language systems, usually to solve small parts of the problem. For example, part-of-speech taggers [1] provide the most likely syntactic category sequence given the word sequence. Probabilities can also be used to choose among a large number of possible syntactic parses for a sentence [2]. But 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. IBM [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.

In speech recognition, we have used statistical methods – notably Hidden Markov Models (HMMs) – to provide a probabilistic framework for the sequence of speech frames. Besides the benefit of having a firm mathematical framework, the HMM paradigm offers three critical features:

1. In principle, as a matter of convention, we consider all answers possible, and find the one that has the highest probability
2. There exist automatic training algorithms to determine the parameters of the model.
3. There exist efficient decoding algorithms for finding the most likely answer.

We have developed what we believed 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.

But first, in order to understand the discussion, we introduce the domain and corpus at this time. We used the ARPA Air Travel Information System (ATIS) domain, which consists of sentences collected from naive users [5, 6]. The users were given complex travel problems to solve with the aid of a spoken language interface to a database of flights and related information. We chose ATIS for three reasons: First, a large corpus of ATIS sentences already exists. Second, ATIS provides an existing evaluation methodology, complete with independent training and test corpora and scoring programs. Third, evaluating on a common corpus makes it easy to compare the performance of the system with those based on different approaches.

In section 2 we describe the overall methodology used in the system. Sections 3 through 6 describe the four major components of the system: Semantic Parsing, Semantic Classification, Discourse Modeling, and Generating Database Queries. Experimental results on the ARPA ATIS task are provided in Section 7.

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 $P(M_d|W, H)$.

However, this problem is quite difficult, since we don't have a single unified mathematical model for the meaning of word sequences in conversations. We commonly factor the problem of establishing meaning into different levels that (we think) we understand better. Then, we adopt an appropriate model for each part of the problem, and use these models in series to perform the whole task.

There is a second advantage for dividing the problem into subproblems. We would not have enough training data (or knowledge in a rule-based system) to learn the problem all at once. But when factored, the number of free parameters and the amount of data needed to learn them become more manageable.

Even when using statistical models, it is essential that our models are appropriate for the task they are asked to perform. There are several aspects of language that we need to capture in our models. Some examples are given below:

- To a first approximation, most of the meaning of a sentence is contained in each of the words used.
- but the order of the words matters.
- Language and meaning is hierarchical, with phrases nested within other phrases.
- Sometimes what you mean (within a domain) is not exactly what you said.
- The meaning of a sentence in a discourse depends on the meanings of the previous sentences, and also on the responses.

We use 4 stages in our model:

1. *Semantic Parse*: 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 $P(T)P(W|T)$.
2. *Semantic Frames*: The semantic trees are not, in themselves, unambiguous meanings. Given the set of candidate semantic parses, T, we find the n-best surface meaning frames, M_s, that maximize $P(M_s, T)P(W|T)$.
3. *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 $P(M_d|H, M_s)P(M_s, T)P(W|T)$.
4. *Backend*: Given the final, unambiguous semantic frame, we produce an SQL expression that carries out the request implied by the meaning.

The first three steps are each estimated completely from annotated data. The fourth step is a compiler that, in principle, should make no mistakes, since it is not trying to make any decisions. Each of these stages is described in the following sections. The Semantic Parse and Semantic Classification components are described more fully in [7].

3. Semantic Parse

The purpose of the semantic parse is to model *how* the user said something. To a first approximation, the parse is syntactic in form, but the labels on the nodes also contain the semantic use of the node.

We compute the probability of the semantic parse, T, given the words, W, using Bayes' rule as:

$$P(T|W) = \frac{P(T)P(W|T)}{P(W)}$$

Since $P(W)$ is constant for any given word string, we rank parses by considering only $P(T)P(W|T)$, which is also the joint probability of $P(T, W)$.

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 [8]. We have incorporated some minor differences. First, we allow all symbols with some probability – even if they have never occurred in training. Second, we use a trigram language model rather than a bigram model. These differences provide a stronger model while being somewhat more robust to lack of training data.

There are special *begin* and *end* states for each nonterminal in addition to the regular symbols to model the probabilities of the first symbol for a nonterminal, and the probability of leaving the nonterminal given the previous symbol. The preterminals also each predict a sequence of symbols, in this case, words. Many words that are used in the same way are grouped together in word classes. For example, all of the names of cities are grouped as a single class, with the probability of each city being used given that the class was used. (The definition of word classes is one piece of information that is not learned. It is relatively simple to define domain-dependent word classes for each domain.)

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.

3.1. Searching the Semantic Parse

We search for the most likely parses using a decoder based on an adaptation of the Earley parsing algorithm [9]. This adaptation, related to that of Stolcke [10] involves reformulating the Earley algorithm to work with probabilistic recursive transition networks rather than with deterministic production rules. For details of the decoder, see [11].

3.2. Training the Parse Probabilities

We train the probabilities on annotated trees. While annotating a large number of sentences with parse trees could be expensive, we have decreased the cost substantially by using a commonly used semiautomatic procedure. First, we enter some parses manually. Then we train the model on those parses. Next, we use the decoder to find the most likely parse for each of the remaining training sentences. A person then

examines these parses to see which ones are right and which are wrong. Even with a small amount of training, the vast majority of the parses are correct, and thus very little effort is required to annotate those. We also correct some of the parses that were recognized incorrectly. Then, we retrain the system using the larger amount of verified training data and again decode the remaining training sentences. At this point, most of the sentences are parsed correctly, and we must only correct those few that were incorrect.

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 as is common in language modeling for speech recognition. We use weights according to the formula in Placeway [12]. We find that backing off to independent semantic and syntactic models results in more precise estimates than just backing off to lower order ngram models.

4. Semantic Classification

Although the semantic parse contains all the elements of the surface meaning, it is not always clear how to put them together. We must create an unambiguous meaning representation. To represent meaning we use a simple nested frame language. Our task, therefore, is to fill the slots of a frame with pieces of meaning from the semantic parse. We treat this as a semantic classification problem. First, we must decide which type of frame we are dealing with. (For example, whether the sentence is about air transportation, ground transportation, or asking about features of different airplanes or the definition of abbreviations.) We choose the frame type by *rescoring* the semantic parse with the same type of ngram language model, but with probability estimates that depend on the particular type of frame, rather than one with frame-independent ngram probabilities.

Next, we consider each concept associated with a preterminal in the semantic parse. Each concept can be used to fill any of the slots in the frame. Many concepts, though, do not fill slots. For example, the preposition "from" in "from Boston" does not itself fill a slot. Rather, when considering what to do with "Boston", it will lead us to fill the ORIGIN slot. Many other words (like "does") will not be used at all. They only helped us to find the semantic parse.

But we do not want to write rules to perform the semantic classification. We use a statistical decision tree as used by [2] to find the probability of each possible action. The decision tree has available to it information about the tree around the preterminal. In particular, it 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.

The resulting frame gives us the surface meaning. Next, we must decide what this sentence means in the context of the preceding discourse.

5. Discourse

Modeling discourse requires quite a different model because it must operate over the sequence of frames. We assume that each meaning frame has two stages: the surface meaning, M_s, derived directly from the sentence (*what* you said), and the discourse meaning, M_d, in context (what you *meant*). Thus, we define a *before-discourse* frame and an *after-discourse* frame. Our goal is to derive the after-discourse frame from a combination of the current before-discourse frame and the sequence of preceding after-discourse frames. We make two assumptions:

1. All of the information needed about the current sentence is represented in its before-discourse frame.
2. All of the discourse context relevant to this utterance is contained in the after-discourse frame of just one preceding *dependent* (not necessarily the most recent) sentence.

Thus, we must decide which of the preceding after-discourse frames provides the context for this sentence. Then, we must decide how that context would modify the frame for the current sentence. We annotated about 4,500 sentences with the before-discourse and after-discourse frames. Again, this process was not as expensive as one might expect. In most cases, the discourse has no effect, so the two frames are identical. In those cases where there was an effect, we used a tool that assumed we inherited all of the fields (constraints and print values) from the dependent sentence that were not contradicted by fields in the current sentence. Then we simply removed those inherited values that were not necessary to produce the final meanings (which were provided with the ATIS corpus).

For the present, we have mainly limited our definition of discourse effects to deciding which of the fields should be inherited from the dependent sentence. For each field we define five possible relations between the current before-discourse frame and the dependent after-discourse frame:

1. *Initialized*: The field was unspecified in the dependent frame, and first appears in the current frame.
2. *Changed*: The field appeared in the preceding frame, and was different in the current frame.
3. *Reiterated*: The field value was given in both frame, but the value is the same.
4. *Tacit*: The field was specified in the dependent frame, but was not specified in the current frame.
5. *Irrelevant*: The field was used in neither frame.

Clearly, the only case of interest is the Tacit case, in which we have to decide whether or not to inherit the field from the dependent sentence. (There are a few exceptions to this, but we won't discuss them here.)

We have experimented with two models for inheritance. The first one (described in [7]) used a decision tree to decide whether each tacit field should be inherited, given the 5-valued features of all the other fields and decisions made so

far. However, this method had too many parameters to be trained on the available data.

The second model, which we use currently, is that modifying (changing or initializing) some fields will cause others to be reset if they haven't been given. For example, if you change the destination of a flight, you probably would not inherit any specification about the date.

5.1. Probability Model

We compute the probability that we would inherit a field given that it is tacit, and given each of the other fields that have been modified or not modified. Using Bayes' rule and an independence assumption, we get

$$P(inherit_i|MOD) \approx P(inherit_i) * \prod_j P(mod_j|inherit_j)$$

where mod_j is the binary value of whether field j has been modified, and $inherit_i$ is the binary value of whether a tacit field i should be inherited (or reset).

We find that this simple model is quite effective at determining which fields should be inherited and which should be reset. We find that, on held out data, only 5.0% of all of the after-discourse frames have one or more incorrect fields, when starting with the correct before discourse frame.

6. Generating Database Queries

Once we have the after-discourse frame, it should be a simple matter to generate the appropriate query to retrieve the desired answer. In principle, the unambiguous meaning should be compiled without error. However, the official rules (the "Principles of Interpretation") for ATIS are quite complicated and require some complex manipulation of the database. A simple frame expression asking us to print flights with two constraints can easily result in a half page of SQL query language.

While this part of the problem may not be as interesting technically, we found that a substantial effort was required to develop the compiler that would generate correct SQL query for the meaning expression. 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

The complete HUM system was trained on 4,500 annotated ATIS2 and ATIS3 utterances and was run - end to end - from words to answer, on the December 1994 test set of the ARPA ATIS (Air Travel Information Service) task.

The error rate was 14.5%. On the "class A" subset of the sentences, which were annotated as context-independent, the error rate was 9.5%. As mentioned above, 3.5% of the answers were wrong because the translation to SQL failed.

8. Summary

We have developed a sentence understanding system that is completely based on learned statistical models. The system uses three levels: 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.

Acknowledgements

This work was supported by the Advanced Research Projects Agency and monitored by Ft. Huachuca under contract No. DABT63-94-C-0061. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

1. Church, K. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," Second Conference on Applied Natural Language Processing, Austin, TX, 1988.
2. "Statistical Decision Tree Models for Parsing," 33rd Annual Meeting of the ACL, Cambridge, MA, 1995, pp. 276-283.
3. Levin, E., and Pieraccini, R., "CHRONUS: The Next Generation," *ARPA Spoken Language Systems Technology Workshop*, Austin, TX, 1995, pp. 269-271.
4. Epstein, M., Papineni, K., Roukos, S., Ward, T., Della Pietra, S., "Statistical Natural Language Understanding Using Hidden Clumpings," *ICASSP96*, Atlanta, GA, May, 1996, pp. 176-179.
5. Bates, M., Boisen, S., and Makhoul, J., "Developing an Evaluation Methodology for Spoken Language Systems.", *ARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June, 1990, pp. 102-108.
6. Price, P., "Evaluation of Spoken Language Systems: the ATIS Domain.", *ARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June, 1990, pp. 91-95.
7. Miller, S., Stallard, D., Bobrow, R., Schwartz, R., "A Fully Statistical Approach to Natural Language Interfaces," *Proceedings of the ACL*, June, 1996.
8. Seneff, S., "TINA: A Natural Language System for Spoken Language Applications," Computational Linguistics, 1992, 18.1, pp. 61-86.
9. Earley, J., "An Efficient Context-Free Parsing Algorithm," Communications of the ACM, 1970, Vol. 6, pp. 451-455.
10. Stolcke, A., "An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities," Computational Linguistics, 1995, 21.2, pp. 165-201.
11. "Hidden Understanding Models," PhD Thesis, Northeastern University, Boston, MA., 1996.
12. Placeway, P., Schwartz, R., Fung, P., and L. Nguyen, "The Estimation of Powerful Language Models from Small and Large Corpora", *ICASSP93*, Minneapolis, MN, April, 1993, pp. 33-36.