A FULLY STATISTICAL APPROACH TO NATURAL LANGUAGE INTERFACES

Scott Miller, David Stallard, Robert Bobrow, Richard Schwartz
BBN Systems and Technologies
70 Fawcett Street
Cambridge, MA 02138

szmiller@bbn.com, stallard@bbn.com, rusty@bbn.com, schwartz@bbn.com

Abstract

We present a natural language interface system which is based entirely on trained statistical models. The system consists of three stages of processing: parsing, semantic interpretation, and discourse. Each of these stages is modeled as a statistical process. The models are fully integrated, resulting in an end-to-end system that maps input utterances into meaning representation frames.

1. Introduction

A recent trend in natural language processing has been toward a greater emphasis on statistical approaches, beginning with the success of statistical part-of-speech tagging programs (Church 1988), and continuing with other work using statistical part-of-speech tagging programs, such as BBN PLUM (Weischedel et al. 1993) and NYU Proteus (Grishman and Sterling 1993). More recently, statistical methods have been applied to domain-specific semantic parsing (Miller et al. 1994), and to the more difficult problem of wide-coverage syntactic parsing (Magerman 1995). Nevertheless, most natural language systems remain primarily rule based, and even systems that do use statistical techniques, such as AT&T Chronus (Levin and Pieraccini 1995), continue to require a significant rule based component. Development of a complete end-to-end statistical understanding system has been the focus of several ongoing research efforts, including (Miller et al. 1995) and (Koppelman et al. 1995). In this paper, we present such a system. The overall structure of our approach is conventional, consisting of a parser, a semantic interpreter, and a discourse module. The implementation and integration of these elements is far less conventional. Within each module, every processing step is assigned a probability value, and very large numbers of alternative theories are pursued in parallel. The individual modules are integrated through an n-best paradigm, in which many theories are passed from one stage to the next, together with their associated probability scores. The meaning of a sentence is determined by taking the highest scoring theory from among the n-best possibilities produced by the final stage in the model.

Some key advantages to statistical modeling techniques are:

- All knowledge required by the system is acquired through training examples, thereby eliminating the need for hand-written rules. In parsing for example, it is sufficient to provide the system with examples specifying the correct parses for a set of training examples. There is no need to specify an exact set of rules or a detailed procedure for producing such parses.
- All decisions made by the system are graded, and there are principled techniques for estimating the gradations. The system is thus free to pursue unusual theories, while remaining aware of the fact that they are unlikely. In the event that a more likely theory exists, then the more likely theory is selected, but if no more likely interpretation can be found, the unlikely interpretation is accepted.

The focus of this work is primarily to extract sufficient information from each utterance to give an appropriate response to a user's request. A variety of problems regarded as standard in computational linguistics, such as quantification, reference and the like, are thus ignored.

To evaluate our approach, we trained an experimental system using data from the Air Travel Information (ATIS) domain (Bates et al. 1990; Price 1990). The selection of ATIS was motivated by three concerns. First, a large corpus of ATIS sentences already exists and is readily available. Second, ATIS provides an existing evaluation methodology, complete with independent training and test corpora, and scoring programs. Finally, evaluating on a common corpus makes it easy to compare the performance of the system with those based on different approaches.

We have evaluated our system on the same blind test sets used in the ARPA evaluations (Pallett et al. 1995), and present a preliminary result at the conclusion of this paper.

The remainder of the paper is divided into four sections, one describing the overall structure of our models, and one for each of the three major components of parsing, semantic interpretation and discourse.

2. Model Structure

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 M_D for the most likely meaning M_0:

$$M_0 = \arg\max_{M_D} P(M_D \mid W, H).$$

Directly modeling $P(M_D \mid W, H)$ 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:

$$M_0 = \arg\max_{M_D} \sum_{M_S,T} P(M_D \mid W,H,M_S,T) P(M_S,T \mid W,H)$$

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,

$$M_0 = \arg\max_{M_D} \sum_{M_S,T} P(M_D \mid H,M_S)\ P(M_S,T \mid W)\ .$$

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

$$P(M_S,T \mid W) = \frac{P(M_S,T)\ P(W \mid M_S,T)}{P(W)},$$

leading to:

$$M_0 = \arg\max_{M_D} \sum_{M_S,T} P(M_D \mid H,M_S) \frac{P(M_S,T)\ P(W \mid M_S,T)}{P(W)}$$

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

$$\sum_{M_S,T} P(M_D \mid H,M_S) \frac{P(M_S,T)\ P(W \mid M_S,T)}{P(W)}$$

is equivalent to finding M_D that maximizes

$$\sum_{M_S,T} P(M_D \mid H,M_S)\ P(M_S,T)\ P(W \mid M_S,T).$$

Thus,

$$M_0 = \arg\max_{M_D} \sum_{M_S,T} P(M_D \mid H,M_S)\ P(M_S,T)\ P(W \mid M_S,T).$$

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:

$$M_0 = \arg\max_{M_D} \sum_{M_S,T} P(M_D \mid H,M_S)\ P(M_S,T)\ P(W \mid T)$$

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:

$$M_0 = \arg\max_{M_D} \left(\max_{M_S,T} \left(P(M_D \mid H,M_S)\ P(M_S,T)\ P(W \mid T) \right) \right)$$

This expression corresponds to the computation actually performed by our system which is shown in Figure 1.

Processing proceeds in three stages:

1. Word string W arrives at the parsing model. The full space of possible parses T is searched for n-best candidates according to the measure $P(T)\ P(W \mid T)$. These parses, together with their probability scores, are passed to the semantic interpretation model.
2. The constrained space of candidate parses T (received from the parsing model), combined with the full space of possible pre-discourse meanings M_S, is searched for n-best candidates according to the measure $P(M_S,T)\ P(W \mid T)$. These pre-discourse meanings, together with their associated probability scores, are passed to the discourse model.

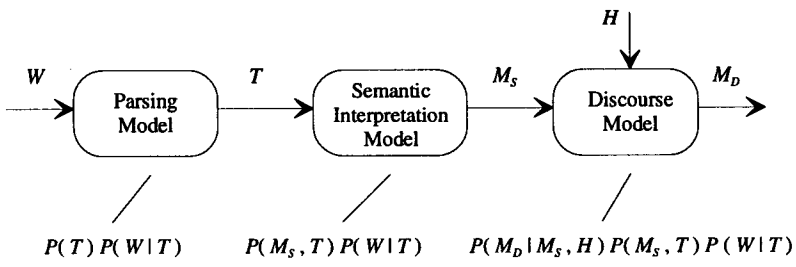

Figure 1: Overview of statistical processing.

3. The constrained space of candidate pre-discourse meanings M_S (received from the semantic interpretation model), combined with the full space of possible post-discourse meanings M_D, is searched for the single candidate that maximizes $P(M_D \mid H, M_S)\ P(M_S, T)\ P(W \mid T)$, conditioned on the current history H. The discourse history is then updated and the post-discourse meaning is returned.

We now proceed to a detailed discussion of each of these three stages, beginning with parsing.

3. Parsing

Our parse representation is essentially syntactic in form, patterned on a simplified head-centered theory of phrase structure. In content, however, the parse trees are as much semantic as syntactic. Specifically, each parse node indicates both a semantic and a syntactic class (excepting a few types that serve purely syntactic functions). Figure 2 shows a sample parse of a typical ATIS sentence. The semantic/syntactic character of this representation offers several advantages:

1. Annotation: Well-founded syntactic principles provide a framework for designing an organized and consistent annotation schema.
2. Decoding: Semantic and syntactic constraints are simultaneously available during the decoding process; the decoder searches for parses that are both syntactically and semantically coherent.
3. Semantic Interpretation: Semantic/syntactic parse trees are immediately useful to the semantic interpretation process: semantic labels identify the basic units of meaning, while syntactic structures help identify relationships between those units.

3.1 Statistical Parsing Model

The parsing model is a probabilistic recursive transition network similar to those described in (Miller et al. 1994) and (Seneff 1992). The probability of a parse tree T given a word string W is rewritten using Bayes rule as:

$$P(T \mid W) = \frac{P(T)\,P(W \mid T)}{P(W)}.$$

Since $P(W)$ is constant for any given word string, candidate parses can be ranked by considering only the product $P(T)\ P(W \mid T)$. The probability $P(T)$ is modeled by state transition probabilities in the recursive transition network, and $P(W \mid T)$ is modeled by word transition probabilities.

- State transition probabilities have the form $P(state_n \mid state_{n-1}, state_{up})$. For example, $P(location/pp \mid arrival/vp\text{-}head, arrival/vp)$ is the probability of a *location/pp* following an *arrival/vp-head* within an *arrival/vp* constituent.
- Word transition probabilities have the form $P(word_n \mid word_{n-1}, tag)$. For example, $P(\text{"class"} \mid \text{"first"}, class\text{-}of\text{-}service/npr)$ is the probability of the word sequence "first class" given the tag *class-of-service/npr*.

Each parse tree T corresponds directly with a path through the recursive transition network. The probability $P(T)\ P(W \mid T)$ is simply the product of each transition

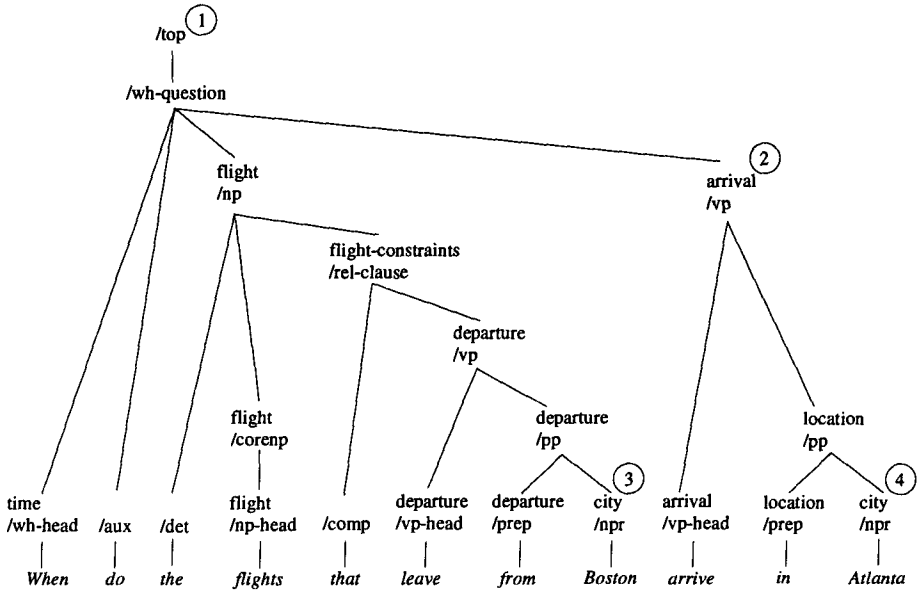

Figure 2: A sample parse tree.

probability along the path corresponding to T.

3.2 Training the Parsing Model

Transition probabilities are estimated directly by observing occurrence and transition frequencies in a training corpus of annotated parse trees. These estimates are then smoothed to overcome sparse data limitations. The semantic/syntactic parse labels, described above, provide a further advantage in terms of smoothing: for cases of undertrained probability estimates, the model backs off to independent syntactic and semantic probabilities as follows:

$$\begin{aligned} P_s(sem/syn_n \mid sem/syn_{n-1}, sem/syn_{up}) = \\ \lambda(sem/syn_n \mid sem/syn_{n-1}, sem/syn_{up}) \\ \times P(sem/syn_n \mid sem/syn_{n-1}, sem/syn_{up}) \\ + (1-\lambda(sem/syn_n \mid sem/syn_{n-1}, sem/syn_{up}) \\ \times P(sem_n \mid sem_{up}) P(syn_n \mid syn_{n-1}, syn_{up}) \end{aligned}$$

where λ is estimated as in (Placeway et al. 1993). Backing off to independent semantic and syntactic probabilities potentially provides more precise estimates than the usual strategy of backing off directly form bigram to unigram models.

3.3 Searching the Parsing Model

In order to explore the space of possible parses efficiently, the parsing model is searched using a decoder based on an adaptation of the Earley parsing algorithm (Earley 1970). This adaptation, related to that of (Stolcke 1995), involves reformulating the Earley algorithm to work with probabilistic recursive transition networks rather than with deterministic production rules. For details of the decoder, see (Miller 1996).

4. Semantic Interpretation

Both pre-discourse and post-discourse meanings in our current system are represented using a simple frame representation. Figure 3 shows a sample semantic frame corresponding to the parse in Figure 2.

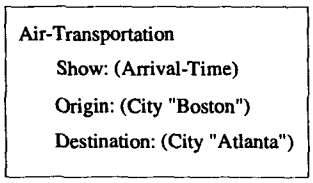

Figure 3: A sample semantic frame.

Recall that the semantic interpreter is required to compute $P(M_S,T)\,P(W \mid T)$. The conditional word probability $P(W \mid T)$ has already been computed during the parsing phase and need not be recomputed. The current problem, then, is to compute the prior probability of meaning M_S and parse T occurring together. Our strategy is to embed the instructions for constructing M_S directly into parse T, resulting in an augmented tree structure. For example, the instructions needed to create the frame shown in Figure 3 are:

1. Create an Air-Transportation frame.
2. Fill the Show slot with Arrival-Time.
3. Fill the Origin slot with (City "Boston")
4. Fill the Destination slot with (City "Atlanta")

These instructions are attached to the parse tree at the points indicated by the circled numbers (see Figure 2). The probability $P(M_S,T)$ is then simply the prior probability of producing the augmented tree structure.

4.1 Statistical Interpretation Model

Meanings M_S are decomposed into two parts: the frame type FT, and the slot fillers S. The frame type is always attached to the topmost node in the augmented parse tree, while the slot filling instructions are attached to nodes lower down in the tree. Except for the topmost node, all parse nodes are required to have some slot filling operation. For nodes that do not directly trigger any slot fill operation, the special operation *null* is attached. The probability $P(M_S, T)$ is then:

$$P(M_S,T) = P(FT,S,T) = P(FT)\,P(T \mid FT)\,P(S \mid FT,T)\,.$$

Obviously, the prior probabilities $P(FT)$ can be obtained directly from the training data. To compute $P(T \mid FT)$, each of the state transitions from the previous parsing model are simply rescored conditioned on the frame type. The new state transition probabilities are:

$$P(state_n \mid state_{n-1}, state_{up}, FT)\,.$$

To compute $P(S \mid FT, T)$, we make the independence assumption that slot filling operations depend only on the frame type, the slot operations already performed, and on the local parse structure around the operation. This local neighborhood consists of the parse node itself, its two left siblings, its two right siblings, and its four immediate ancestors. Further, the syntactic and semantic components of these nodes are considered independently. Under these assumptions, the probability of a slot fill operation is:

$$\begin{aligned} P(slot_n \mid FT, S_{n-1}, sem_{n-2}, \ldots, sem_n, \ldots, sem_{n+2}, \\ syn_{n-2}, \ldots, syn_n, \ldots, syn_{n+2}, \\ sem_{up1}, \ldots, sem_{up4}, syn_{up1}, \ldots, syn_{up4}) \end{aligned}$$

and the probability $P(S \mid FT, T)$ is simply the product of all such slot fill operations in the augmented tree.

4.2 Training the Semantic Interpretation Model

Transition probabilities are estimated from a training corpus of augmented trees. Unlike probabilities in the parsing model, there obviously is not sufficient training data to estimate slot fill probabilities directly. Instead, these probabilities are estimated by statistical decision trees similar

to those used in the Spatter parser (Magerman 1995). Unlike more common decision tree classifiers, which simply classify sets of conditions, statistical decision trees give a probability distribution over all possible outcomes. Statistical decision trees are constructed in a two phase process. In the first phase, a decision tree is constructed in the standard fashion using entropy reduction to guide the construction process. This phase is the same as for classifier models, and the distributions at the leaves are often extremely sharp, sometimes consisting of one outcome with probability 1, and all others with probability 0. In the second phase, these distributions are smoothed by mixing together distributions of various nodes in the decision tree. As in (Magerman 1995), mixture weights are determined by deleted interpolation on a separate block of training data.

4.3 Searching the Semantic Interpretation Model

Searching the interpretation model proceeds in two phases. In the first phase, every parse T received from the parsing model is rescored for every possible frame type, computing $P(T \mid FT)$ (our current model includes only a half dozen different types, so this computation is tractable). Each of these theories is combined with the corresponding prior probability $P(FT)$ yielding $P(FT)\ P(T \mid FT)$. The n-best of these theories are then passed to the second phase of the interpretation process. This phase searches the space of slot filling operations using a simple beam search procedure. For each combination of FT and T, the beam search procedure considers all possible combinations of fill operations, while pruning partial theories that fall beneath the threshold imposed by the beam limit. The surviving theories are then combined with the conditional word probabilities $P(W \mid T)$, computed during the parsing model. The final result of these steps is the n-best set of candidate pre-discourse meanings, scored according to the measure $P(M_S,T)\ P(W \mid T)$.

5. Discourse Processing

The discourse module computes the most probable post-discourse meaning of an utterance from its pre-discourse meaning and the discourse history, according to the measure:

$$P(M_D \mid H, M_S)\ P(M_S,T)\ P(W \mid T).$$

Because pronouns can usually be ignored in the ATIS domain, our work does not treat the problem of pronominal reference. Our probability model is instead shaped by the key discourse problem of the ATIS domain, which is the inheritance of constraints from context. This inheritance phenomenon, similar in spirit to one-anaphora, is illustrated by the following dialog::

USER1: I want to fly from Boston to Denver.

SYSTEM1: <displays Boston to Denver flights>

USER2: Which flights are available on Tuesday?

SYSTEM2: <displays Boston to Denver flights for Tuesday>

In USER2, it is obvious from context that the user is asking about flights whose ORIGIN is BOSTON and whose DESTINATION is DENVER, and not all flights between any two cities. Constraints are not always inherited, however. For example, in the following continuation of this dialogue:

USER3: Show me return flights from Denver to Boston,

it is intuitively much less likely that the user means the "on Tuesday" constraint to continue to apply.

The discourse history H simply consists of the list of all post-discourse frame representations for all previous utterances in the current session with the system. These frames are the source of candidate constraints to be inherited. For most utterances, we make the simplifying assumption that we need only look at the last (i.e. most recent) frame in this list, which we call M_P.

5.1 Statistical Discourse Model

The statistical discourse model maps a 23 element input vector X onto a 23 element output vector Y. These vectors have the following interpretations:

- X represents the combination of previous meaning M_P and the pre-discourse meaning M_S.
- Y represents the post-discourse meaning M_D.

Thus,

$$P(M_D \mid H, M_S) = P(Y \mid X).$$

The 23 elements in vectors X and Y correspond to the 23 possible slots in the frame schema. Each element in X can have one of five values, specifying the relationship between the filler of the corresponding slot in M_P and M_S:

INITIAL - slot filled in M_S but not in M_P

TACIT - slot filled in M_P but not in M_S

REITERATE - slot filled in both M_P and M_S; value the same

CHANGE - slot filled in both M_P and M_S; value different

IRRELEVANT - slot not filled in either M_P or M_S

Output vector Y is constructed by directly copying all fields from input vector X except those labeled TACIT. These direct copying operations are assigned probability 1. For fields labeled TACIT, the corresponding field in Y is filled with either INHERITED or NOT-INHERITED. The probability of each of these operations is determined by a statistical decision tree model. The discourse model contains 23 such statistical decision trees, one for each slot position. An ordering is imposed on the set of frame slots, such that inheritance decisions for slots higher in the order are conditioned on the decisions for slots lower in the order.

The probability $P(Y \mid X)$ is then the product of all 23 decision probabilities:

$$P(Y \mid X) = P(y_1 \mid X)\ P(y_2 \mid X, y_1) \ldots P(y_{23} \mid X, y_1, y_2, \ldots, y_{22})\ .$$

5.2 Training the Discourse Model

The discourse model is trained from a corpus annotated with both pre-discourse and post-discourse semantic frames. Corresponding pairs of input and output (X, Y) vectors are computed from these annotations, which are then used to train the 23 statistical decision trees. The training procedure for estimating these decision tree models is similar to that used for training the semantic interpretation model.

5.3 Searching The Discourse Model

Searching the discourse model begins by selecting a meaning frame M_P from the history stack H, and combining it with each pre-discourse meaning M_S received from the semantic interpretation model. This process yields a set of candidate input vectors X. Then, for each vector X, a search process exhaustively constructs and scores all possible output vectors Y according to the measure $P(Y \mid X)$ (this computation is feasible because the number of TACIT fields is normally small). These scores are combined with the pre-discourse scores $P(M_S,T)\ P(W \mid T)$, already computed by the semantic interpretation process. This computation yields:

$$P(Y \mid X)\ P(M_S,T)\ P(W \mid T)\ ,$$

which is equivalent to:

$$P(M_D \mid H, M_S)\ P(M_S,T)\ P(W \mid T)\ .$$

The highest scoring theory is then selected, and a straightforward computation derives the final meaning frame M_D from output vector Y.

6. Experimental Results

We have trained and evaluated the system on a common corpus of utterances collected from naive users in the ATIS domain. In this test, the system was trained on approximately 4000 ATIS 2 and ATIS 3 sentences, and then evaluated on the December 1994 test material (which was held aside as a blind test set). The combined system produced an error rate of 21.6%. Work on the system is ongoing, however, and interested parties are encouraged to contact the authors for more recent results.

7. Conclusion

We have presented a fully trained statistical natural language interface system, with separate models corresponding to the classical processing steps of parsing, semantic interpretation and discourse. Much work remains to be done in order to refine the statistical modeling techniques, and to extend the statistical models to additional linguistic phenomena such as quantification and anaphora resolution.

8. Acknowledgments

We wish to thank Robert Ingria for his effort in supervising the annotation of the training corpus, and for his helpful technical suggestions.

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 Ft. Huachuca under Contract Nos. DABT63-94-C-0061 and DABT63-94-C-0063 . The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

9. References

Bates, M., Boisen, S., and Makhoul, J. "Developing an Evaluation Methodology for Spoken Language Systems." Speech and Natural Language Workshop, Hidden Valley, Pennsylvania, 102-108.

Church, K. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text." Second Conference on Applied Natural Language Processing, Austin, Texas.

Earley, J. (1970). "An Efficient Context-Free Parsing Algorithm." Communications of the ACM, 6, 451-455.

Grishman, R., and Sterling, J. "Description of the Proteus System as Used for MUC-5." Fifth Message Understanding Conference, Baltimore, Maryland, 181-194.

Koppelman, J., Pietra, S. D., Epstein, M., Roukos, S., and Ward, T. "A statistical approach to language modeling for the ATIS task." Eurospeech 1995, Madrid.

Levin, E., and Pieraccini, R. "CHRONUS: The Next Generation." Spoken Language Systems Technology Workshop, Austin, Texas, 269-271.

Magerman, D. "Statistical Decision Tree Models for Parsing." 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge, Massachusetts, 276-283.

Miller, S. (1996). "Hidden Understanding Models," Northeastern University, Boston, MA.

Miller, S., Bates, M., Bobrow, R., Ingria, R., Makhoul, J., and Schwartz, R. "Recent Progress in Hidden Understanding Models." Spoken Language Systems Technology Workshop, Austin, Texas, 276-280.

Miller, S., Bobrow, R., Ingria, R., and Schwartz, R. "Hidden Understanding Models of Natural Language." 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 25-32.

Pallett, D., Fiscus, J., Fisher, W., Garofolo, J., Lund, B., Martin, A., and Przybocki, M. "1994 Benchmark Tests for the ARPA Spoken Language Program." Spoken Language Systems Technology Workshop, Austin, Texas.

Placeway, P., Schwartz, R., Fung, P., and Nguyen, L. "The Estimation of Powerful Language Models from Small and Large Corpora." IEEE ICASSP, 33-36.

Price, P. "Evaluation of Spoken Language Systems: the ATIS Domain." Speech and Natural Language Workshop, Hidden Valley, Pennsylvania, 91-95.

Seneff, S. (1992). "TINA: A Natural Language System for Spoken Language Applications." Computational Linguistics, 18,1, 61-86.

Stolcke, A. (1995). "An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilites." Computational Linguistics, 21(2), 165-201.

Weischedel, R., Ayuso, D., Boisen, S., Fox, H., Ingria, R., Matsukawa, T., Papageorgiou, C., MacLaughlin, D., Kitagawa, M., Sakai, T., Abe, J., Hosihi, H., Miyamoto, Y., and Miller, S. "Description of the PLUM System as Used for MUC-5." Fifth Message Understanding Conference, Baltimore, Maryland, 93-107.