Towards Understanding Text with a Very Large Vocabulary

Damaris Ayuso, R. Bobrow, Dawn MacLaughlin, Marie Meteer, Lance Ramshaw, Rich Schwartz, Ralph Weischedel

BBN Systems and Technologies Corporation
10 Moulton St.
Cambridge, MA 02138

1. Introduction

In order to meet the information processing demands of the next decade, natural language systems must have the capability of processing very large amounts of text, commonly called "messages", from highly diverse sources written in any of a few dozen languages. One of the key issues in building systems with this scale of competence is handling large numbers of different words and word senses. Natural language understanding systems today are typically limited to vocabularies of less than 10,000 words; tomorrow's systems will need vocabularies at least 5 times that to effectively handle the volume and diversity of messages needing to be processed.

One method of handling large vocabularies is simply increasing the size of the lexicon. Research efforts at IBM [Chodorow, et al. 1988; Neff, et al. 1989], Bell Labs [Church, et al. 1989], New Mexico State University [Wilks 1987], and elsewhere have used mechanical processing of on-line dictionaries to infer at least minimal syntactic and semantic information from dictionary definitions. However, even assuming a very large lexicon already exists, it can never be complete. Systems aiming for coverage of unrestricted language in broad domains must continually deal with new words and novel word senses.

Systems with very large lexicons have the additional problems of an exploding search space, of disambiguating multiple syntactic and semantic possibilities when full interpretations are possible, and of combining partial interpretations into something meaningful when a full interpretation is not found. For instance, in *The Wall Street Journal*, the average sentence length is 21 words, more than twice the average sentence length of the corpus for the Air Travel Information System used in spoken language systems research. If the worst case complexity of a parser is n^3, then the search space can be eight times worse than in spoken language interfaces.

A key element of our approach to these problems is the use of probabilistic models to control the greatly increased search space inherent in large vocabularies. We have observed that the state of the art in natural language processing (NLP) today is analogous to that in speech processing roughly prior to 1980, when purely knowledge-based approaches required much detailed, hand-crafted knowledge from several sources (e.g., acoustic, phonetic, etc.). Speech systems then, like NLP systems today, were brittle, required much hand-crafting, were limited in accuracy, and were not scalable. A revolution in speech technology has occurred since 1980, when probabilistic models were incorporated into the control structure for combining multiple sources of knowledge (providing improved accuracy and increased scalability) and as algorithms for training the system on large bodies ("corpora") of data were applied (providing reduced cost in moving the technology to a new application domain).

We are exploring the use of probabilistic models and training in NLP in a new pilot study, whose overall goal is to increase the robustness, precision, and scalability of natural language understanding systems. In the initial phase of the study, we are addressing issues raised by the huge vocabularies in open texts. We are experimenting with a variety of techniques for disambiguating word uses, selecting syntactic interpretations, and acquiring information about new words--techniques that can be applied both when a word is initially encountered and in handling the word more effectively the next time it is encountered.

This paper reports the results of the first three months of this new effort. We have applied techniques from speech processing, such as "tri-tag" models and probability models on context-free grammars. We report on our initial experiments in using tri-tag models for hypothesizing parts of speech, as well as new results on the size of the corpus needed for training these models, and their use in processing unknown words. We discuss our use of a context-free probabilistic language model to help in selecting the correct parse from among multiple parses. Finally, we present a preliminary approach to the problem of learning the lexical syntax of new words in context and using our probabilistic language model to aid in selecting the interpretation to learn from.

2. Probabilistic Part of Speech Models

One straightforward way to use probabilities is in assigning parts of speech to words. Models predicting part of speech can serve to cut down the search space a parser must consider in processing known words and can be used as one input to more complex strategies for inferring lexical and semantic information about unknown words. We have explored the use of such models in both contexts.

Simple but powerful models to predict part of speech can be derived using a corpus that has been tagged (or labelled) as to part of speech [Church 1988; de Marken 1990]. Using a tagged corpus to train the model is called "supervised training", since a human has prepared the correct training data. This is in contrast to "unsupervised training" where the process is fully automated. For example, in unsupervised part of speech tagging, one can use a corpus without annotation for training, a dictionary that lists parts of speech for the most frequently occurring words, and an initial probability assignment, e.g., a uniform probability distribution or probability estimates from a previous, related domain. An iterative procedure then revises the probability estimates so as to maximize the probability over the whole corpus.

Our supervised training experiments used a *tri-tag model* based on a corpus from the University of Pennsylvania consisting of *Wall Street Journal* articles in which each word or punctuation mark has been tagged with one of 47 parts of speech, as shown in the following example:

Terms/NNS were/VBD not/RB disclosed/VBN ./.

A tri-tag model predicts the relative likelihood of a particular tag given the two preceding tags, e.g. how likely is the tag RB on the third word in the above example, given that the two previous words were tagged NNS and VBD. Using the UPenn corpus, we counted for each possible pair of tags, the number of times that the pair was followed by each possible third tag, and then derived from those counts a probabilistic tri-tag model. We also estimated from the training data the conditional probability of each particular word given a known tag (e.g., how likely is the work to be "terms" if the tag is NNS); this is called the "word emit" probability. Both of these probability estimates used *padding* to an arbitrary estimate to avoid setting the probability for unseen tri-tags or unseen word senses to zero.

Given these probabilities, one can then predict the maximum-likelihood tag sequence for a given word sequence. Using the tri-tag probabilities, we computed the probabilities of all possible paths in the tag space through the sentence, selected the path whose overall probability was highest, and then took the tag predictions from that path. We replicated the result [Church 1988] that this process is able to predict the parts of speech with only a 3-5% error rate when the possible parts of speech of the words are known. We believe that this error rate could be reduced still further and extend the success to unknown words.

Using the UPenn set of parts of speech, unknown words can be in any of the 22 open-class parts of speech. The tri-tag model can be used to estimate the most probable one. While random choice among the 22 open classes would be expected to show an error rate for new words of 91.5%, our initial results using the model showed an error rate of only 51.6%. The best previously reported error rate was 75% [Kuhn & de Mori 1990]. Note that the error rate should be reduced even further by using more knowledge, such as capitalization knowledge and morphology.

While supervised training is shown here to be very effective, it requires a correctly tagged corpus. We have done some experiments to quantify how much tagged data is really necessary, and to suggest ways to handle new words when using such models.

In these experiments, we demonstrated that the training set can, in fact, be much smaller than might have been expected. One rule of thumb suggests that the training set needs to be large enough to contain 10 instances of each type of tag sequence in order for their probabilities to be estimated with reasonable accuracy. This would imply that a tri-tag model using 47 possible parts of speech would need a bit more than a million words of training. However, we found that much less training data was necessary, as illustrated in Figure 1.

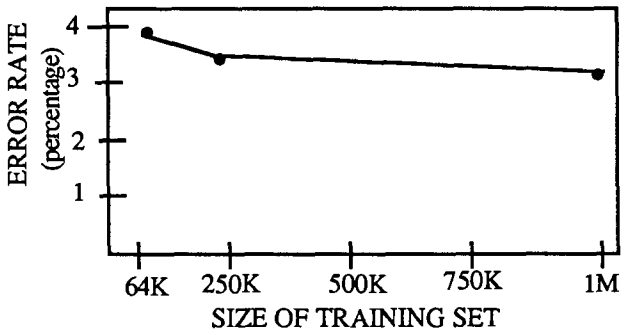

Figure 1: Size of Tri-tag Training Sets

In our experiments, the error rate for a supervised tri-tag model increased only from 3.30% to 3.87% when the size of the training set was reduced from 1 million words to 64,000 words. This is probably because most of the possible tri-tag sequences never actually appear. All that is really necessary, recalling the rule of thumb, is enough training to allow for 10 of each of the tag sequences that do occur. There were 16,170 unique triples in our training set, so the rule of thumb would suggest that 160,000 words would be sufficient training. This would explain why the degradation in performance was slight when the size of the corpus was reduced. The benefits of probabilistic modeling therefore seem applicable to new tag sets, subdomains, or languages without needing prohibitively large corpora.

3. Probabilistic Language Model

Probabilities can also quantify the likelihoods of alternative complete interpretations of a sentence. In these experiments, we used the grammar of the Delphi component from BBN's HARC system [Stallard 1989], which combines syntax and semantics in a unification formalism. We developed a *context-free* model, which estimates the probability of each rule in the grammar independently (in contrast to a context-sensitive model, such as the tri-tag model described above, which bases the probability of a tag on what other tags are in the adjacent context).

In our context-free model, we associate a probability with each rule of the grammar. For each distinct major category (left-hand side) of the grammar, there is a set of context-free rules

$$LHS \leftarrow RHS_1$$
$$LHS \leftarrow RHS_2$$
$$\bullet\bullet\bullet$$
$$LHS \leftarrow RHS_n.$$

For each rule, we estimate the probability of the right-hand side given the left-hand side.

The probability of a syntactic structure S, given the input string W, is then modelled by the product of the probabilities of the rules used in S. ([Chitrao & Grishman 1990] used a similar context-free model.) Using this model, we explored the following issues:

- What method of training the rule probabilities should be employed?
- How much (little) training data is required for reliable estimates?
- How is system performance impacted?
- Do the results suggest refinements in the probability model?

Our intention is to use the Treebank corpus being developed at the University of Pennsylvania as a source of correct structures for training. However, until that material becomes available, we have run initial experiments using small training sets taken from an existing question-answering corpus of sentences about a personnel database. To our surprise, we found that as little as 100 sentences of supervised training (in which a person, using graphical tools, identifies the correct parse) is sufficient to improve the ranking of the interpretations found. In our tests, the NLP system produces all interpretations satisfying all syntactic and semantic constraints. From that set, the intended interpretation must be chosen. The context-free probability model reduced the error rate on an independent test set by a factor of two to four, compared to random selection from the interpretations satisfying all knowledge-based constraints.

We tested the predictive power of rule probabilities using this model both in unsupervised and in supervised mode. In the former case, the input is all parse trees (whether correct or not) for the sentences in the training set. In the latter case, the training data included a specification of the correct parse as hand picked by the grammar's author from among the parse trees produced by the system.

The detailed results from using a training set of 81 sentences appear in the histogram in Figure 2.

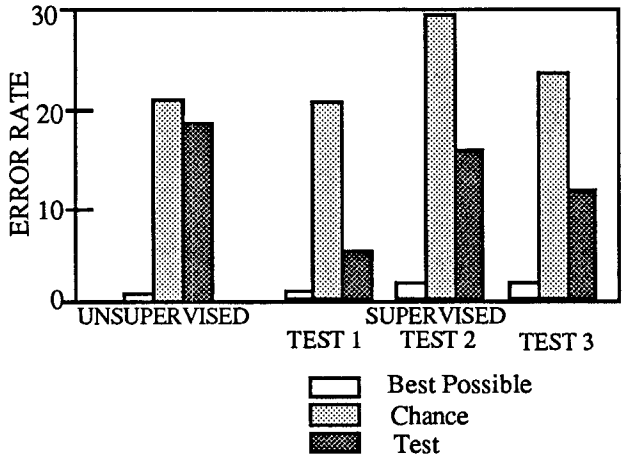

Figure 2: Predictions of Probabilistic Language Model

The "best possible" error rates for each test indicates the percentage of cases for which none of the interpretations produced by the system was judged correct, so that no selection scheme could achieve a lower error rate than that. The "chance" score gives the error rate that would be expected with random selection from all interpretations produced. The "test" column shows the error rate with the supervised or unsupervised probability model in question. The first supervised test had an 81.4% improvement, and the second a 50.8% improvement, and the third a 56% improvement. These results state how much better than chance the given model did as a percentage of the maximum possible improvement.

We expect to improve the model's performance by recording probabilities for other features in addition to just the set of rules involved in producing them. For example, in the grammar used for this test, two different attachments for a prepositional phrase produced trees with the same set of rules, but differing in shape. Thus the simple, context-free model based on the product of rule probabilities could not capture preferences concerning such attachment. By adding to the model probabilities for such additional features, we expect that the power of the probabilistic model to

automatically select the correct parse can be substantially increased.

4. Learning Lexical Syntax

One purpose for probabilistic models is to contribute to handling new words or partially understood sentences. We have done preliminary experiments that show that there is promise in learning lexical syntactic and semantic features from context when probabilistic tools are used to help control the ambiguity.

In our experiments, we used a corpus of sentences each with one word that the system did not know. To create the corpus, we began with a corpus of sentences known to parse from the personnel question-answering domain (our goal, again, is to use the Treebank data from the University of Pennsylvania for such training when it becomes available). We then replaced one word in each sentence with an undefined word.

For example, in the following sentence, the word "contact" is undefined in the system: *Who in Division Four is the contact for MIT*? That word has both a noun and a verb part of speech; however, the pattern of parts of speech of the words surrounding "contact" causes the tri-tag model to return a high probability that the word is a noun. Using unification variables for all possible features of a noun, the parser produces multiple parses. Applying the context-free rule probabilities to select the most probable of the resulting parses allows the system to conclude both syntactic and semantic facts about "contact". Syntactically, the system discovers that it is a count noun, with third person singular agreement. Semantically, the system learns (from the use of who) that contact is in the semantic class PERSONS.

Furthermore, the partially-specified semantic representation for the sentence as a whole also shows the semantic relation to SCHOOLS, which is expressed here by the *for* phrase. Thus, even a single use of an unknown word in context can supply useful data about its syntactic and semantic features.

Probalistic modelling plays a key role in this process. While context sensitive techniques for inferring lexical features can contribute a great deal, they can still leave substantial ambiguity. As a simple example, suppose the word "list" is undefined in the sentence "List the employees." The tri-tag model predicts both a noun and a verb part of speech in that position. Using an underspecified noun sense combined with the usual definitions for the rest of the words yields no parses. However, an underspecified verb sense yields three parses, differing in the subcategorization frame of the verb "list". For more complex sentences, even with this very limited protocol, the number of parses for the appropriate word sense can reach into the hundreds.

Using the rule probabilities acquired through supervised training (described in the previous section), the likelihood of the ambiguous interpretations resulting from a sentence with an unknown word was computed. Then we tested whether the tree ranked most highly matched the tree previously selected by a person as the correct one. This tree equivalence test was based on the trees' structure and on the rule applied at each node; while an underspecified tree might have some less-specified feature values than the chosen fully-specified tree, it would still be equivalent in the sense above.

Of 160 inputs with an unknown word, in 130 cases the most likely tree matched the correct one, for an error rate of 18.75%, while picking at random would have resulted in an error rate of 63.14%, for an improvement by better than a factor of 3. This suggests that probabilistic modeling can be a powerful tool for controlling the high degree of ambiguity in efforts to automatically acquire lexical data.

We have also begun to explore heuristics for combining lexical data for a single word acquired from a number of partial parses. There are some cases in which the best approach is to unify the two learned sets of lexical features, so that the derived sense becomes the sum of the information learned from the two examples. For instance, the verb subcategorization information learned from one example could be thus combined with agreement information learned from another. On the other hand, there are many cases, including alternative subcategorization frames, where each of the encountered options needs to be included as separate alternatives.

5. Conclusions

In trying to address the problems inherent in understanding text using very large vocabularies, we found that the use of probabilistic models was crucial in obtaining useful results. The three main problems addressed by this paper were (1) reducing ambiguity resulting from multiple parts of speech, (2) reducing parse ambiguity, and (3) learning lexical information of new words encountered in the text.

Using supervised training for tri-tag probabilities, we achieved a 3-5% error rate on a test set in picking the correct part of speech. Our experiments showed that a smaller training set than previously expected (64,000 words rather than 1 million) was needed in order to achieve a good level of performance.

For reducing interpretation ambiguity, our context-free probability model, trained in supervised mode on only 81 sentences, was able to reduce the error rate for selecting the correct parse on independent test sets by a factor of 2-4.

For the problem of processing new words in the text, the tri-tag model reduced the error rate for picking the correct part of speech for such words from 91.5% to 51.6%. And once the possible parts of speech for a word are known (or

hypothesized using the tri-tag model), the probabilistic language model proved useful in indicating which parses (obtained using the unknown word) should be looked at for learning more complex lexical information about the word.

6. Future Work

We plan to explore ways in which to reduce the error rates resulting from our current models. For example, the potential of using a weighted combination of n-tag models for a range of n, as opposed to a single tri-tag model, can be studied. We also plan to use a more complex probabilistic model of grammar, one that more realistically represents the biases in language, for example, by using conditional probabilities relying on more than just one level of context-free rules.

In a different direction, we plan to explore automatic methods for learning semantic information. We will explore the use of the Common Facts Database (CFDB) [Crowther 1989], which was derived from an on-line dictionary with a base vocabulary of 65K words. The CFDB would be useful in assigning semantic classes to noun phrases, for example, as well as in providing information on the classes of verb arguments.

Acknowledgements

The work reported here was supported by the Advanced Research Projects Agency and was monitored by the Rome Air Development Center under Contract No. F30602-87-D-0093. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

References

Chitrao, M.V. and Grishman, R. Statistical Parsing of Messages. *Proceedings of the Speech and Natural Language Workshop* June 1990.

Chodorow, M.S., Ravin, Y., and Sachar, H.E. A Tool for Investigating the Synonymy Relation in a Sense Disambiguated Thesaurus. ACL Proceedings of the Second Conference on Applied Natural Language Processing 1988, 144-152.

Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. Proceedings of the Second Conference on Applied Natural Language Processing, ACL, 1988, 136-143.

Church, K., Gale, W.A., Hanks, P., and Hindle, D. Parsing, Word Associations and Typical Predicate-Argument Relations. *Proceedings of the Speech and Natural Language Workshop* Oct. 1989, 75-81.

Crowther, W. A Common Facts Database. *Proceedings of the Speech and Natural Language Workshop* Feb. 1989, 89-93.

de Marcken, C.G. Parsing the LOB Corpus. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics* 1990, 243-251.

Kuhn, R., and De Mori, R. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990), 570-583.

Neff, M.S., and Boguraev, B.K. Dictionaries, Dictionary Grammars, and Dictionary Parsing. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics* 1989, 91-101.

Stallard, D. Unification-Based Semantic Interpretation in the BBN Spoken Language System. *Proceedings of the Speech and Natural Language Workshop* Oct. 1989, 39-46.

Wilks, Y. A Tractable Machine Dictionary as a Resource for Computational Semantics. *NAIC 1987 Natural Language Planning Workshop* 1987, 97-123.