Bootstrapping Statistical Processing into a Rule-based Natural Language Parser

Stephen D. Richardson
Microsoft Research
One Microsoft Way
Redmond, WA 98052

Abstract

This paper describes a "bootstrapping" method which uses a broad-coverage, rule-based parser to compute probabilities while parsing an untagged corpus of NL text, and which then incorporates those probabilities into the processing of the same parser as it analyzes new text. Results are reported which show that this method can significantly improve the speed and accuracy of the parser without requiring the use of annotated corpora or human-supervised training during the computation of probabilities.

1 Introduction

For decades, the majority of NL parsers have been "rule-based." In such parsers, knowledge about the syntactic structure of a language is written in the form of linguistic rules, and these rules are applied by the parser to input text segments in order to produce the resulting parse trees. Information about individual words, such as what parts-of-speech they may be, is usually stored in an online dictionary, or "lexicon," which is accessed by the parser for each word in the input text prior to applying the linguistic rules.

Although rule-based parsers are widely-used in real, working NLP systems, they have the disadvantage that extensive amounts of (dictionary) data and labor (to write the rules) by highly-skilled linguists are required in order to create, enhance, and maintain them. This is especially true if the parser is required to have "broad coverage", i.e., if it is to be able to parse NL text from many different domains (what one might call "general" text).

In the last few years, there has been increasing activity in the computational linguistics community focused on making use of statistical methods to acquire information from large corpora of NL text, and on using that information in statistical NL parsers. Instead of being stored in the traditional form of dictionary data and grammatical rules, linguistic knowledge in these parsers is represented as statistical parameters, or probabilities. These probabilities are commonly used together with simpler, less specified, dictionary data and/or rules, thereby taking the place of much of the information created by skilled labor in rule-based systems.

Advantages of the statistical approach that are claimed by its proponents include a significant decrease in the amount of rule coding required to create a parser that performs adequately, and the ability to "tune" a parser to a particular type of text simply by extracting statistical information from the same type of text. Perhaps the most significant disadvantage appears to be the requirement for large amounts of training data, often in the form of large NL text corpora that have been annotated with hand-coded tags specifying parts-of-speech, syntactic function, etc. There have been a number of efforts to extract information from corpora that are not tagged (e.g., Kupiec and Maxwell 1992), but the depth of information thus obtained and its utility in "automatically" creating a NL parser is usually limited.

To overcome the need for augmenting corpora with tags in order to obtain more useful information, researchers in statistical NLP have experimented with a variety of strategies, some of which employ varying degrees of traditional linguistic abstraction. Su and Chang (1992) group words in untagged corpora into equivalence classes, according to their possible parts-of-speech. They then perform statistical analyses over these equivalence classes, rather than over the words themselves, in order to obtain higher-level trigram language models that will be used later by their statistics-based parser. Brown et al. (1992) have similarly resorted to reducing inflected word forms to their underlying lemmas before estimation of statistical parameters. Briscoe and Carroll (1993) carry the use of traditional rule-based linguistics a step further by using a unification-based grammar as a starting point. Through a process of human-supervised training on a small corpus of text, a statistical model is then developed which is used to rank the parses produced by the grammar for a given input. A similar method of interactive training has been used by Simmons and Yu (1991) to produce favorable results.

Beyond the realm of simply using traditional linguistics to enhance the quality of data extracted from corpora by statistical methods, there have been attempts to create hybrid systems that incorporate statistical information into already well-developed rule-based frameworks. For example, McKee and Maloney (1992) have used common statistical methods to extract information such as part-of-speech frequency, verb subcategorization frames, and prepositional phrase attachment preferences from corpora and have then incorporated it into the processing in their knowledge-based parser in order to quickly expand its coverage in new domains.

In comparing rule-based approaches with those which are more purely statistics-based, and including everything in between, one could claim that there is some constant amount of linguistic knowledge that is required to create an NL parser, and one must either code it explicitly into the parser (using rules), or use statistical methods to extract it from sources such as text corpora. Furthermore, in the latter case, the extraction of useful information from the raw data in corpora is facilitated by additional information provided through manual tagging, through "seeding" the process with linguistic abstractions (e.g., parts-of-speech), or through the interaction of human supervisors during the extraction process. In any case, it appears that in addition to information that may be obtained by statistical methods, generalized linguistic knowledge from a human source is also clearly desirable, if not required, in order to create truly capable parsers.

Proponents of statistical methods usually point to the data-driven aspect of their approach as enabling them to create robust parsers that can handle "real text." Although many rule-based parsers have been limited in scope, we believe that it is indeed possible to create and maintain broad-coverage, rule-based NL systems (e.g., Jensen 1993), by carefully studying and using ample amounts of data to refine those systems. It has been our experience that the complexity and difficulty of creating such rule-based systems can be readily managed if one has a powerful and comprehensive set of tools. Nevertheless, it is also clearly desirable to be able to use statistical methods to adapt (or tune) rule-based systems automatically for particular types of text as well as to acquire additional linguistic information from corpora and to integrate it with information that has been developed by trained linguists.

To the end of incorporating statistics-based processing into a rule-based parser, we have devised a "bootstrapping" method. This method uses a rule-based parser to compute part-of-speech and rule probabilities while processing a large, non-annotated corpus. These probabilities are then incorporated into the very same parser, thereby providing guidance to the parser as it assigns parts of speech to words and applies rules during the processing of new text.

Although our method relies on the existence of a broad-coverage, rule-based parser, which, as discussed at the beginning of this paper, is not trivial to develop, the benefits of this approach are that relevant statistical information can be obtained automatically from large untagged corpora, and that this information can be used to improve significantly the speed and accuracy of the parser. This method also obviates the need for any human-supervised training during the parsing process and allows for "tuning" the parser to particular types of text.

2 The Bootstrapping Method

We use a broad-coverage, rule-based, bottom-up, chart parser as the basis for this work. It utilizes the Microsoft English Grammar (MEG), which is a set of augmented phrase structure grammar rules containing conditions designed to eliminate many potential, but less-preferred, parses. It seeks to produce a single approximate syntactic parse for each input, although it may also produce multiple parses or even a "fitted" parse in the event that a well-formed parse is not obtained. The "approximate" nature of a parse is exemplified by the packing of many attachment ambiguities, where phrases often default to simple right attachment and a notation is made for further processing to resolve the ambiguity at a later point in the NLP system.

The bootstrapping method begins by using the rule-based parser to parse a large corpus of untagged NL text. During parsing, frequencies that will be used to compute rule and part-of-speech probabilities are obtained. For rule probabilities, these frequencies in their simplest form include the number of times that each rule r creates a node n_r in a well-formed parse tree and the total number of times that r was attempted (i.e., the sequence of constituents $c_1,\ldots,c_m$ that trigger r occurred in the chart and r's conditions were evaluated relative to those constituents). At the end of parsing the corpus, the former frequency is divided by the latter frequency to obtain the probability for each rule, as given in Figure 1 below. The reason for using the denominator as given rather than the number of times $c_1,\ldots,c_m$ occurs below n_r in a parse tree is that it adjusts for the conditions on rules contained in MEG, which may allow many such sequences of constituents to occur in the chart, but only very few of them to occur in the final parse tree. In this case, the probability of a rule might be skewed in favor of trying it more often than it should be, unless the denominator were based on constituents in the chart vs. in the parse tree.

$$P(r|c_1,\ldots,c_m) = \frac{(\#\text{ times } n_r \text{ occurs in trees})}{(\#\text{ times } c_1,\ldots,c_m \text{ occur in chart})}$$

Figure 1. Simple rule probability

For part-of-speech probabilities, the frequencies obtained during parsing include the number of times a word w occurs having a particular part-of-speech p in a well-formed parse tree and the total number of times that w occurs. Once again, at the end of parsing, the former frequency is divided by the latter to obtain the simple probability that a word will occur with a particular part of speech, as given in Figure 2.

$$P(p|w) = \frac{(\#\text{ times } w \text{ occurs having } p \text{ in trees})}{(\#\text{ times } w \text{ occurs in trees})}$$

Figure 2. Simple part-of-speech probability

Since the choice was made to use the denominator for rule probabilities given above, the part-of-speech probabilities must be normalized so that the two sets of probabilities are compatible and may be used together during the probabilistic algorithm described below. The normalization is achieved by multiplying each part-of-speech probability by the ratio of the average probability of all the rules over the average probability of all the parts of speech for all the words. This effectively lowers the part-of-speech probabilities into the same range as the rule probabilities, so that as the probabilistic algorithm proceeds, it will try lower probability parts of speech for words at a consistent point relative to the application of lower probability rules.

After computing and normalizing the probabilities, they are incorporated into the same rule-based parser used to compute them. The parser is guided by these probabilities, while parsing any new input, to seek the most probable path through the parse search space, instead of taking the "all-paths" breadth-first approach it took when parsing without the use of the probabilities. A simplified description of the chart parsing algorithm as guided by probabilities is given in Figure 3 below. The term *record* used in the algorithm may be likened to an *edge* in traditional chart parsing terminology. A *part-of-speech record* refers to an edge representing one (of possibly many) of the parts of speech for a given word. A list (*PLIST* below) of potential rule applications and part-of-speech records, sorted by probability in descending order (i.e., highest probability first), is maintained throughout the execution of the algorithm. The *next most probable* potential rule application or part-of-speech record is always located at the top of PLIST.

1. Put all of the part-of-speech records for each word in the input into PLIST, forcing the probability of the highest probability part-of-speech record for each word to 1 (ensuring that at least one part-of-speech record for each word will be put into the chart immediately).
2. Process the next most probable item in PLIST:
   a. If it is a potential rule application, remove it from PLIST and try the rule. If the rule succeeds, add a record representing a new sub-tree to the chart.
   b. Otherwise, if it is a part-of-speech record, remove it from PLIST and add it directly to the chart.
3. If a record was added to the chart in step 2, identify all new potential rule applications (by examining the constituent sequences in the chart), obtain their probabilities (from those that were computed and stored previously), and put them in their appropriate position in PLIST.
4. Stop if a record representing a parse tree for the entire input string was generated or if PLIST is empty, otherwise go to step 2.

Figure 3. Probability-directed chart parsing algorithm

The PLIST in this algorithm is similar to the ordered agenda used in the "best first" parser described by Allen (1994). However, in contrast to Allen's parser, the probabilities used by this algorithm do not take into account the probabilities of the underlying nodes in each subtree, which in the former case are multiplied together (on the basis of a pragmatically motivated independence assumption) to obtain a probability representative of the entire subtree. Therefore, this algorithm is not guaranteed to produce the most probable parse first. In practice, though, the algorithm does achieve good results and avoids having to deal with the problems that Allen admits are encountered when trying to apply a best-first strategy based on independence assumptions to a large-scale grammar. These include a rapid drop-off of the probabilities as subtrees grow deeper, causing a regression to nearly breadth-first searching. We desire instead to maintain parsing efficiency at the cost of potentially not generating some number of most probable parses, while still generating a large number of those that are most probable. The results reported below appear to bear this out.

3 Discussion

One potential disadvantage of the bootstrapping method is that the parser can reinforce its own bad behavior. However, this may be controlled by parsing a large amount of data,[1] and then by using only the probabilities computed for "shorter" sentences (currently, those less than 35 words) for which a single, well-formed parse is obtained (in contrast to those for which multiple or "fitted" parses are obtained). Our assessment thus far is that our parser generates straightforward structures for the large majority of such sentences, resulting in fairly accurate rule and part-of-speech probabilities. In many ways, this strategy is similar to the strategies employed by Hindle and Rooth (1993) and by Kinoshita et al. (1993) in that we rely on processing of less ambiguous data to provide information to assist the parser in processing the more difficult, ambiguous cases.

Another factor in avoiding the reinforcement of bad behavior is our linguist's skill in making sure that the most common structures parse accurately. As

[1] We have used the 1 million word Brown corpus to compute our current set of statistics, but anticipate using larger corpora.

we evaluate the output of the probabilistic version of our parser, our linguist continues, in a principled manner, to add and change conditions on rules to correct problems with parse structures and parts-of-speech. We have just made changes to the parser that enable it to use one set of probabilities (along with the changes our linguist made on that base) during parsing while computing another set. This will allow us to iterate during the development of the parser in a rule-based/statistics-based cycle, and to experiment with the effects of one set of methods on the other.

Also, the simple probabilities described in the previous section are only a starting point. Already, we have dependently conditioned the probabilities of rules on the following characteristics of the parse tree nodes generated by them:

1. *l*, the length (in words) of the text covered by the node, divided by 5
2. *d*, the distance (in words) that the text covered by the node is from the end of the sentence, divided by 5
3. *m*, the minimal path length (in nodes) of the node

The division of the first two conditioning factors by 5 serves to lump together the values obtained during probability computation, thereby decreasing the potential for sparse data. The third factor, the minimal path length of a node, is defined as the smallest number of contiguous nodes that covers the text between the node and the end of the sentence, where nodes are contiguous if the text strings they represent are contiguous in the sentence. The rule probability computation, including these three conditioning factors, is given in Figure 4. The term "composite" in the denominator means that the specific l_i, d_i, and m_i are computed as if the constituents $c_1,\ldots,c_m$ were one node.

Although these conditioning factors are not linguistically motivated in the theoretical sense, they have nevertheless contributed significantly to further improving the speed and accuracy of the parser. Results based on their use are provided in the next section. They were identified based on an inspection of the conditions in the MEG rules and how those rules go about building up well-formed parse tree structures (namely, right to left between certain clause and phrase boundaries). Through experimentation, it was confirmed that these three factors are all helpful in guiding the parser to explore the most probable linguistic structures in the search space in an order that is consistent with how the MEG rules tend to build these structures. Specifically, MEG tends to extend structures from right to left that are longer and span from any given word to the end of a clause, especially to the end of the sentence. The advantageous use of these conditions points to the importance of carefully considering various aspects of the existing rule set when integrating statistical processing within a rule-based parser.

$$P\left(r \middle| c_1,\ldots,c_m,l_i,d_i,m_i\right) = \frac{(\#\text{ times } n_r \text{ with } l_i,d_i,\text{ and } m_i \text{ occurs in trees})}{\left(\begin{array}{l}\#\text{ times } c_1,\ldots,c_m \text{ with composite } l_i,d_i,\text{ and } m_i\\ \text{occur in chart}\end{array}\right)}$$

Figure 4. Conditioned rule probability

In the future, we anticipate conditioning the probabilities further based on truly linguistic considerations, such as the rule history or head word of a given structure. This has been suggested in works such as Black, et al. (1993). We also anticipate experimenting cautiously with various independence assumptions in order to decrease our parameter space as we increase the number of conditioning factors. In all of these endeavors, we will seek to determine the most beneficial interplay between the rule-based and statistics-based aspects of our system.

4 Results

We have used our parser to compute both simple and conditioned probabilities, as described above, during the parsing of the 1 million word Brown corpus. In round numbers, this process took about 34 hours on a 486/66 PC, for an average of 2.5 seconds per sentence. There are about 55,000 sentences in the Brown corpus, averaging 18 words in length, but those over 35 words in length (more than 7,000) were not parsed, for the reasons given earlier.

The probabilities thus computed were incorporated for use by the probabilistic algorithm of the parser, and the parser was then applied to two sets of selected sentences in order to evaluate the anticipated improvements in parsing speed and accuracy. The first set contained 500 sentences, averaging 17 words in length, and randomly selected from different sources, including articles from Time magazine and the Wall Street Journal, linguistic textbook examples, and correspondence. The efficiency of the parser in processing these sentences, both with and without probabilities, is documented in Table 1.

	Average records in chart	Average rules attempted	Average parsing time (secs)
No probabilities	364	12836	2.416
Simple probabilities	238	6633	1.879
Conditioned probabilities	181	2367	1.231

Table 1. Comparison of parsing efficiency over 500 sentences

Useful measures of parsing efficiency include the total number of records in the chart when a parse is obtained or the parser otherwise stops, the number of rules attempted for a given input string and, of course, the time required to parse a string (assuming a dedicated, non-multi-tasking computer system). On average, using the conditioned probabilities resulted in half as many records being placed in the chart during the processing of a sentence and a corresponding speed-up by a factor of 2. Rule attempts decreased by more than a factor of 5. A large number of sentences parsed many times faster than with the non-probabilistic algorithm, but this was tempered in the averaging process by a number of long sentences that parsed in nearly the same time, and on very rare occasions, slightly slower.[2]

In the probabilistic algorithm used in this evaluation, we also implemented a low-probability cutoff to stop the parser from continuing to apply rules after a certain number of rules (whose probability is less than the average probability of all the rules) had been attempted. This number is multiplied by the number of words in a sentence (to adjust for the obvious fact that more rule applications are needed for longer sentences) and has been determined experimentally by running the parser on sets of sentences and examining how often a well-formed (in contrast to "fitted") parse is actually obtained after a certain number of less-than-average rules have been attempted. The parser currently produces a fitted parse for just over 20% of the sentences in the first set described above. In practice, using this low-probability cutoff rarely increases the number of fitted parses obtained, and then only slightly (perhaps a percentage point or so). This is more than offset by the use of the probabilities which, due to their positive effect on parsing efficiency, allow for the successful parsing of much longer and more complicated sentences without exhausting computational resources such as available computer memory.

The second set of sentences on which the parser was evaluated contained 100 sentences, roughly half being randomly selected from a linguistic textbook and the other half from some Time magazine articles. Although the former half were fairly short (10 words/sentence), they exhibited a variety of linguistic structures, in contrast to the somewhat more straightforward, but longer (17 words/sentence), sentences from the latter half. All the sentences in this set shared the characteristic that the parser produced two or more parses for each of them. The parse trees produced by the parser for these sentences were examined and it was determined whether the correct parse

[2] Slower parsing is actually possible, when the probabilities turn out to be useless for a given sentence, because of the overhead of maintaining and accessing the PLIST described in Figure 3.

was produced first by the probabilistic algorithm, using both simple and conditioned probabilities. For the non-probabilistic algorithm, the parse trees were ordered according to the degree of right attachment they exhibited (i.e., deepest structures first). As shown in Table 2, the algorithm using conditioned probabilities selected the correct parse more than twice as often as simple right attachment. It is interesting to note that while the probabilistic algorithm performed somewhat better on the shorter textbook sentences than on the longer magazine sentences, right attachment performed worse. This is most likely due to the wide variety of (not simple right-branching) linguistic structures in the textbook sentences.

	Linguistic textbook sentences (46)	Time magazine sentences (54)	Total (100)
No probabilities (ordered by degree of right attachment)	33%	43%	38%
Simple probabilities	74%	67%	70%
Conditioned probabilities	89%	76%	82%

Table 2. Comparison of correct parse selection over 100 sentences for which multiple parses are produced

5 Conclusion

We have described a “bootstrapping” method, which uses a broad-coverage, rule-based parser to compute probabilities while parsing a non-annotated corpus of NL text, and which incorporates those probabilities into the very same parser for use in analyzing new text. The results reported from an evaluation of this method show that it can significantly improve the speed and accuracy of the parser. A salient feature of this method is that it does not require the use of annotated corpora or human-supervised training during the computation of the probabilities.

References

Allen, J. 1994. *Natural Language Understanding,* 2nd Edition, ch. 7. New York: Benjamin/Cummings.

Black, E., F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos. 1993. Towards history-based grammars: using richer models for probabilistic parsing. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics,* 31-37.

Briscoe, T., and J. Carroll. 1993. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics,* 19, no. 1:25-59.

Brown, P., S. Della Pietra, V. Della Pietra, J. Lafferty, and R. Mercer. 1992. Analysis, statistical transfer, and synthesis in machine translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation* (Montreal, Canada), 83-98.

Hindle, D., and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics,* 19, no. 1:103-120.

Jensen, K. 1993. PEG: the PLNLP English grammar. In *Natural Language Processing: the PLNLP Approach,* ed. K. Jensen, G. Heidorn, and S. Richardson, 29-45. Boston: Kluwer Academic Publishers.

Kinoshita, S., M. Shimazu, and H. Hirakawa. 1993. Better translation with knowledge extracted from the source text. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation* (Kyoto, Japan), 240-251.

Kupiec, J., and J. Maxwell. 1992. Training stochastic grammars from unlabelled text corpora. In *AAAI-92 Workshop Program on Statistically-Based NLP Techniques* (San Jose, CA), 14-19.

McKee, D., and J. Maloney. 1992. Using statistics gained from corpora in a knowledge-based NLP system. In *AAAI-92 Workshop Program on Statistically-Based NLP Techniques* (San Jose, CA), 81-89.

Simmons, R., and Y. Yu. 1991. The acquisition and application of context sensitive grammar for English. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 122-129.

Su, K., and J. Chang. 1992. Why corpus-based statistics-oriented machine translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation* (Montreal, Canada), 249-262.