Experience with an Easily Computed Metric for Ranking Alternative Parses

George E. Heidorn

Computer Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Abstract

This brief paper, which is itself an extended abstract for a forthcoming paper, describes a metric that can be easily computed during either bottom-up or top-down construction of a parse tree for ranking the desirability of alternative parses. In its simplest form, the metric tends to prefer trees in which constituents are pushed as far down as possible, but by appropriate modification of a constant in the formula other behavior can be obtained also. This paper includes an introduction to the EPISTLE system being developed at IBM Research and a discussion of the results of using this metric with that system.

Introduction

Heidorn (1976) described a technique for computing a number for each node during the *bottom-up* construction of a parse tree, such that a node with a smaller number is to be preferred to a node with a larger number covering the same portion of text. At the time, this scheme was used primarily to select among competing noun phrases in queries to a program explanation system. Although it appeared to work well, it was not extensively tested. Recently, as part of our research on the EPISTLE system, this idea has been modified and extended to work over entire sentences and to provide for *top-down* computation. Also, we have done an analysis of 80 sentences with multiple parses from our data base to evaluate the performance of this metric, and have found that it is producing very good results.

This brief paper, which is actually an extended abstract for a forthcoming paper, begins with an introduction to the EPISTLE system, to set the stage for the current application of this metric. Then the metric's computation is described, followed by a discussion of the results of the 80-sentence analysis. Finally, some comparisons are made to related work by others.

The EPISTLE System

In its current form, the EPISTLE system (Miller, Heidorn and Jensen 1981) is intended to do critiquing of a writer's use of English in business correspondence, and can do some amount of grammar and style checking. The central component of the system is a parser for assigning grammatical structures to input sentences. This is done with NLP, a LISP-based natural language processing system which uses augmented phrase structure grammar (APSG) rules (Heidorn 1975) to specify how text is to be converted into a network of nodes consisting of attribute-value pairs and how such a network can be converted into text. The first process, *decoding*, is done in a bottom-up, parallel processing fashion, and the inverse process, *encoding*, is done in a top-down, serial manner. In the current application the network which is constructed is simply a decorated parse tree, rather than a meaning representation.

Because EPISTLE must deal with unrestricted input (both in terms of vocabulary and syntactic constructions), we are trying to see how far we can get initially with almost no semantic information. In particular, our information about words is pretty much limited to parts-of-speech that come from an on-line version of a standard dictionary of over 100,000 entries, and the conditions in our 250 decoding rules are based primarily on syntactic cues. We strive for what we call a unique *approximate* parse for each sentence, a parse that is not necessarily semantically accurate (e.g., prepositional phrase attachments are not always done right) but one which is adequate for the text critiquing tasks, nevertheless.

One of the things we do periodically to test the performance of our parsing component is to run it on a set of 400 actual business letters, consisting of almost 2,300 sentences which range in length up to 63 words, averaging 19 words per sentence. In two recent runs of this data base, the following results were obtained:

No. of parses	June 1981	Dec. 1981
0	57%	36%
1	31%	41%
2	6%	11%
>2	6%	12%

The improvement in performance from June to December can be attributed both to writing additional grammar rules and to relaxing overly restrictive conditions in other rules. It can be seen that this not only had the desirable effect of reducing the percentage of no-parse sentences (from 57% to 36%) and increasing the percentage of single-parse sentences (from 31% to 41%), but it also had the undesirable side effect of increasing the multiple-parse sentences (from 12% to 23%). Because we expect this situation to continue as we further increase our grammatical coverage, the need for a method of ranking multiple parses in order to select the best one on which to base our grammar and style critiques is acutely felt.

The Metric and Its Computation

The metric can be stated by the following recursive formula:

$$\text{Score}_{\text{Phrase}} = \sum_{\text{Mods}} K_{\text{Mod}}(\text{Score}_{\text{Mod}}+1)$$

where the *lowest* score is considered to be the *best*. This formula says that the score associated with a phrase is equal to the sum of the scores of the modifying phrases of that phrase adjusted in a particular way, namely that the score of each modifier is increased by 1 and then multiplied by a constant K appropriate for that type of modifier. A phrase with no modifiers, such as an individual word, has a score of 0. This metric is based on a *flat* view of syntactic structure which says that each phrase consists of a head word and zero or more pre- and post-modifying phrases. (In this view a sentence is just a big verb phrase, with modifiers such as subject, objects, adverbs, and subordinate clauses.)

In its simplest form this metric can be considered to be nothing more than the numerical realization of Kimball's Principle Number Two (Kimball 1972): "Terminal symbols optimally associate to the lowest nonterminal node." (Although Kimball calls this principle *right association* and illustrates it with right-branching examples, it can often apply equally well to left-branching structures.) One way to achieve this simplest form is to use a K of 0.1 for all types of modifiers.

An example of the application of the metric in this simplest form is given in Figure 1. Two parse trees are shown for the sentence, "See the man with the telescope," with a score attached to each node (other than those that are zero). A node marked with an asterisk is the head of its respective phrase. In this form of flat parse tree a prepositional phrase is displayed as a noun phrase with the preposition as an additional premodifier. As an example of the calculation, the score of the PP here is computed as 0.1(0+1)+0.1(0+1), because the scores of its modifiers — the ADJ and the PREP — are each 0. Similarly, the score of the NP in the second parse tree is computed as 0.1(0+1)+0.1(0.2+1), where the 0.2 within it is the score of the PP.

It can be seen from the example that in this simplest form the individual digits of the score after the decimal point tell how many modifiers appear at each level in the phrase (as long as there are no more than nine modifiers at any level). The farther down in the parse tree a constituent is pushed, the farther to the right in the final score its contribution will appear. Hence, a deeper structure will tend to have a smaller score than a shallower structure, and, therefore, be preferred. In the example, this is the second tree, with a score of 0.122 vs. 0.23. That is not to say that this would be the semantically correct tree for this sentence in all contexts, but only that if a choice cannot be made on any other grounds, this tree is to be preferred.

Applying the metric in its simplest form does not produce the desired result for all grammatical constructions, so that values for K other than 0.1 must be used for some types of modifiers. It basically boils down to a system of *rewards* and *penalties* to make the metric reflect preferences determined heuristically. For example, the preference that a potential auxiliary verb is to be used as an auxiliary rather than as a main verb when both parses are possible can be realized by using a K of 0, a reward, when picking up an auxiliary verb. Similarly, a K of 2, a penalty, can be used to increase the score (thereby lessening the preference) when attaching an adverbial phrase as a premodifier in a lower level clause (rather than as a postmodifier in a higher level clause). When semantic information is available, it can be used to select appropriate values for K, too, such as using 100 for an anomalous combination.

Straightforward application of the formula given above implies that the computation of the score can be done in a bottom-up fashion, as the modifiers of each phrase are picked up. However, it can also be done in a top-down manner after doing a little bit of algebra on the formula to expand it and regroup the terms. In the EPISTLE application it is the latter approach that is being used. There is actually a set of ten NLP *encoding* rules that do the computation in a downward traversal of a completed parse tree, determining the appropriate constant to use at each node. The top-down method of computation could be done during top-down parsing of the sort typically used with ATN's, also.

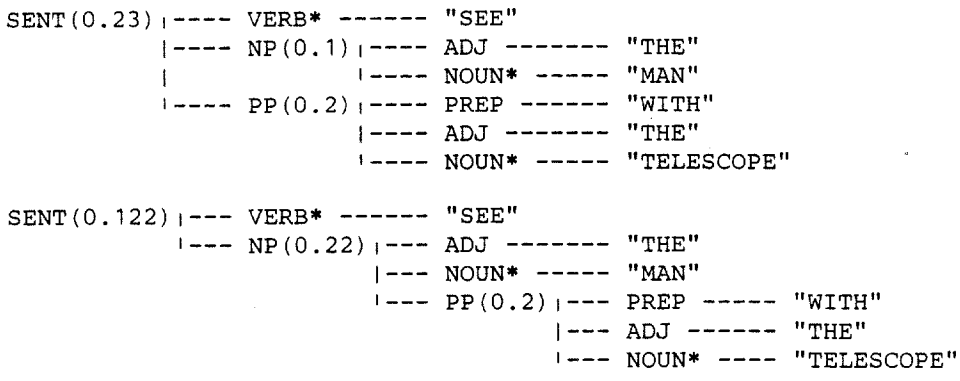

Figure 1. Two alternative parses with their scores.

Performance of the Metric

To test the performance of the metric in our EPISTLE application, the parse trees of 80 multiple-parse sentences were analyzed to determine if the metric favored what we considered to be the best tree for our purposes. A raw calculation said it was right in 65% of the cases. However, further analysis of those cases where it was wrong showed that in half of them the parse that it favored was one which will not even be produced when we further refine our grammar rules. If we eliminate these from consideration, our success rate increases to 80%. Out of the remaining "failures," more than half are cases where semantic information is required to make the correct choice, and our system simply does not yet have enough such information to deal with these. The others, about 7%, will require further tuning of the constant K in the formula. (In fact, they all seem to involve VP conjunction, for which the metric has not been tuned at all yet.)

The analysis just described was based on multiple parses of order 2 through 6. Another analysis was done separately on the double parses (i.e. order 2). The results were similar, but with an adjusted success rate of 85%, and with almost all of the remainder due to the need for more semantic information.

It is also of interest to note that significant right-branching occurred in about 75% of the cases for which the metric selected the best parse. Most of these were situations in which the grammar rules would allow a constituent to be attached at more than one level, but simply pushing it down to the lowest possible level with the metric turned out to produce the best parse.

Related Research

There has not been much in the literature about using numerical scores to rank alternative analyses of segments of text. One notable exception to this is the work at SRI (e.g., Paxton 1975 and Robinson 1975, 1980), where *factor statements* may be attached to an APSG rule to aid in the calculation of a score for a phrase formed by applying the rule. The score of a phrase is intended to express the likelihood that the phrase is a correct interpretation of the input. These scores apparently can be integers in the range 0 to 100 or symbols such as GOOD or POOR. This method of scoring phrases provides more flexibility than the metric of this paper, but also puts more of a burden on the grammar writer.

Another place in which scoring played an important role is the syntactic component of the BBN SPEECHLIS system (Bates 1976), where an integer score is assigned to each *configuration* during the processing of a sentence to reflect the likelihood that the path which terminates on that configuration is correct. The grammar writer must assign weights to each arc of the ATN grammar, but the rest of the computation appears to be done by the system, utilizing such information as the number of words in a constituent. Although this scoring mechanism worked very well for its intended purpose, it may not be more generally applicable.

A very specialized scoring scheme was used in the JIMMY3 system (Maxwell and Tuggle 1977), where each parse network is given an integer score calculated by rewarding the finding of the actor, object, modifiers, and prepositional phrases and punishing the ignoring of words and terms. Finally, there is Wilks' counting of dependencies to find the analysis with the greatest *semantic density* in his Preference Semantics work (eg., Wilks 1975). Neither of these purports to propose scoring methods that are more generally applicable, either.

Acknowledgements

I would like to thank Karen Jensen, Martin Chodorow and Lance Miller for the help that they have given me in the development and testing of this parsing metric, and John Sowa for his comments on an earlier draft of this paper.

References

Bates, M. 1976. "Syntax in Automatic Speech Understanding" *Am. J. Comp. Ling.* Microfiche 45.

Heidorn, G.E. 1975. "Augmented Phrase Structure Grammars" *Theoretical Issues in Natural Language Processing,* B.L. Webber and R.C. Schank (Eds.), Assoc. for Comp. Ling., June 1975, 1-5.

Heidorn, G.E. 1976. "An Easily Computed Metric for Ranking Alternative Parses" *Presented at the Fourteenth Annual Meeting of the Assoc. for Comp. Ling.,* San Francisco, October 1976.

Kimball, J. 1972. "Seven Principles of Surface Structure Parsing in Natural Language" *Cognition 2, 1,* 15-47.

Maxwell, B.D. and F.D. Tuggle 1977. "Toward a 'Natural' Language Question-Answering Facility" *Am. J. Comp. Ling.* Microfiche 61.

Miller, L.A., G.E. Heidorn and K. Jensen 1981. "Text-Critiquing with the EPISTLE System: An Author's Aid to Better Syntax" *AFIPS - Conference Proceedings,* Vol. 50, May 1981, 649-655.

Paxton, W.H. 1975. "The Definition System" in *Speech Understanding Research,* SRI Annual Technical Report, June 1975, 20-25.

Robinson, J.J. 1975. "A Tuneable Performance Grammar" *Am. J. Comp. Ling.,* Microfiche 34, 19-33.

Robinson, J.J. 1980. "DIAGRAM: A Grammar for Dialogues" *SRI Technical Note 205,* Feb. 1980.

Wilks, Y. 1975. "An Intelligent Analyzer and Understander of English" *Comm. ACM 18, 5* (May 1975), 264-274.