Parsing a Natural Language Using Mutual Information Statistics*

David M. Magerman and **Mitchell P. Marcus**
CIS Department
University of Pennsylvania
Philadelphia, PA 19104
E-mail: magerman@linc.cis.upenn.edu

Abstract

The purpose of this paper is to characterize a constituent boundary parsing algorithm, using an information-theoretic measure called generalized mutual information, which serves as an alternative to traditional grammar-based parsing methods. This method is based on the hypothesis that constituent boundaries can be extracted from a given sentence (or word sequence) by analyzing the mutual information values of the part-of-speech n-grams within the sentence. This hypothesis is supported by the performance of an implementation of this parsing algorithm which determines a recursive unlabeled bracketing of unrestricted English text with a relatively low error rate. This paper derives the generalized mutual information statistic, describes the parsing algorithm, and presents results and sample output from the parser.

Introduction

A standard approach to parsing a natural language is to characterize the language using a set of rules, a grammar. A grammar-based parsing algorithm recursively determines a sequence of applications of these rules which reduces the sentence to a single category. Besides determining sentence structure, grammar-based approaches can also identify attributes of phrases, such as case, tense, and number, and they are known to be extremely effective at characterizing and classifying sentences. But these techniques are generally demonstrated using only a subset of the grammar of the language. In order for a grammar-based parser to be applied to unrestricted natural language text, it must account for most of the complexities of the natural language. Thus, one must first concisely describe the bulk of the grammar of that language, an extremely difficult task.

This characterization suggests that a solution to the problem of parsing unrestricted natural language text must rely on an alternative to the grammar-based approach. The approach presented in this paper is based on viewing part-of-speech sequences as stochastic events and applying probabilistic models to these events. Our hypothesis is that constituent boundaries, or "distituents," can be extracted from a sequence of n categories, or an n-gram, by analyzing the mutual information values of the part-of-speech sequences within that n-gram. In particular, we will demonstrate that the generalized mutual information statistic, an extension of the bigram (pairwise) mutual information of two events into n-space, acts as a viable measure of continuity in a sentence.

One notable attribute of our algorithm is that it actually includes a grammar — a distituent grammar, to be precise. A distituent grammar is a list of tag pairs which *cannot* be adjacent within a constituent. For instance, *noun prep* is a known distituent in English, since the grammar of English does not allow a constituent consisting of a noun followed by a preposition. Notice that the nominal head of a noun phrase may be followed by a prepositional phrase; in the context of distituent parsing, once a sequence of tags, such as *(prep noun)*, is grouped as a constituent, it is considered as a unit.

Based on our claim, mutual information *should* detect distituents without aid, and a distituent grammar should not be necessary. However, the application of mutual information to natural language parsing depends on a crucial assumption about constituents in a natural language. Given any constituent n-gram, $a_1a_2\ldots a_n$, the probability of that constituent occur-

*This work was partially supported by DARPA grant No. N0014-85-K0018, by DARPA and AFOSR jointly under grant No. AFOSR-90-0066, and by ARO grant No. DAAL 03-89-C0031 PRI. Special thanks to Ken Church, Stuart Shieber, Max Mintz, Beatrice Santorini, and Tom Veatch for their valued input, guidance and support.

ring is usually significantly higher than the probability of $a_1 a_2 \ldots a_n a_{n+1}$ occurring. This is true, in general, because most constituents appear in a variety of contexts. Once a constituent is detected, it is usually very difficult to predict what part-of-speech will come next. While this assumption is not valid in every case, it turns out that a handful of cases in which it is invalid are responsible for a majority of the errors made by the parser. It is in these few cases that we appeal to the distituent grammar to prevent these errors.

The distituent parsing algorithm is an example of a stochastic, corpus-based approach to parsing. In the past, a significant disadvantage of probabilistic parsing techniques has been that these methods were prone to higher than acceptable error rates. By contrast, the mutual information parsing method presented in this paper is based on a statistic which is both highly accurate and, in the cases where it is inaccurate, highly consistent. Taking advantage of these two attributes, the generalized mutual information statistic and the distituent grammar combine to parse sentences with, on average, two errors per sentence for sentences of 15 words or less, and five errors per sentence for sentences of 30 words or less (based on sentences from a reserved test subset of the Tagged Brown Corpus, see footnote 1). Many of the errors on longer sentences result from conjunctions, which are traditionally troublesome for grammar-based algorithms as well. Further, this parsing technique is extremely efficient, parsing a 35,000 word corpus in under 10 minutes on a Sun 4/280.

It should be noted at this point that, while many stochastic approaches to natural language processing that utilize frequencies to estimate probabilities suffer from sparse data, *sparse data is not a concern in the domain of our algorithm.* Sparse data usually results from the infrequency of *word* sequences in a corpus. The statistics extracted from our training corpus are based on tag n-grams for a set of 64 tags, not word n-grams.[1] The corpus size is sufficiently large that enough tag n-grams occur with sufficient frequency to permit accurate estimates of their probabilities. Therefore, the kinds of estimation methods of $(n+1)$-gram probabilities using n-gram probabilities discussed in Katz (1987) and Church & Gale (1989) are not needed.

This line of research was motivated by a series of successful applications of mutual information statistics to other problems in natural language processing. In the last decade, research in speech recognition (Jelinek 1985), noun classification (Hindle 1988), predicate argument relations (Church & Hanks 1989), and other areas have shown that mutual information statistics provide a wealth of information for solving these problems.

[1]The corpus we use to train our parser is the Tagged Brown Corpus (Francis and Kucera, 1982). Ninety percent of the corpus is used for training the parser, and the other ten percent is used for testing. The tag set used is a subset of the Brown Corpus tag set.

Mutual Information Statistics

Before discussing the mutual information parsing algorithm, we will demonstrate the mathematical basis for using mutual information statistics to locate constituent boundaries. Terminology becomes very important at this point, since there are actually two statistics which are associated with the term "mutual information," the second being an extension of the first.

In his treatise on information theory, *Transmission of Information* (Fano 1961), Fano discusses the mutual information statistic as a measure of the interdependence of two signals in a message. This bigram mutual information is a function of the probabilities of the two events:

$$\mathcal{MI}(x,y) = \log \frac{\mathcal{P}_{X,Y}(x,y)}{\mathcal{P}_X(x)\mathcal{P}_Y(y)}. \qquad (1)$$

Consider these events not as signals but as parts-of-speech in sequence in a sentence. Then an estimate of the mutual information of two categories, xy, is:

$$\mathcal{MI}(x,y) \approx \log \frac{\frac{\#\ xy \text{ in corpus}}{\text{total } \# \text{ of bigrams in corpus}}}{\left(\frac{\#\ x}{\text{corpus size}}\right)\left(\frac{\#\ y}{\text{corpus size}}\right)}. \qquad (2)$$

In order to take advantage of context in determining distituents in a sentence, however, one cannot restrict oneself to looking at pairs of tokens, or bigrams; one must be able to consider n-grams as well, where n spans more than one constituent. To satisfy this condition, we can simply extend mutual information from bigrams to n-grams by allowing the events x and y to be part-of-speech n-grams instead of single parts-of-speech. We will show that this extension is not sufficient for the task at hand.

The second statistic associated with mutual information is what we will call "generalized mutual information," because it is a generalization of the mutual information of part-of-speech bigrams into n-space. Generalized mutual information uses the context on both sides of adjacent parts-of-speech to determine a measure of its distituency in a given sentence. We will discuss this measure below.

While our distituent parsing technique relies on generalized mutual information of n-grams, the foundations of the technique will be illustrated with the base case of simple mutual information over the space of bigrams for expository convenience.

Notation

Before continuing with a mathematical derivation of the generalized mutual information statistic, some mathematical and statistical notation should be explained.

Many different probability functions will be referred to in this paper. P_Ω represents a probability function which maps the set Ω onto the interval $[0,1]$. In equation 1, reference is made to three different probability functions: $\mathcal{P}_X$, $\mathcal{P}_Y$, and $\mathcal{P}_{X,Y}$. The subscripts of these functions indicate their domains, X, Y, and $X \times Y$, respectively. However, these subscripts will be omitted from the remaining equations, since the domain of each probability function will be indicated by its arguments.

The subscripts and superscripts of the mutual information functions can also be somewhat confusing. The bigram mutual information function, denoted as $\mathcal{MI}$, maps the cross-product of two event spaces onto the real numbers. $\mathcal{MI}_n$ is a vector-valued function indicating the mutual information of any two parts of an n-gram, $x_1 \dots x_n$. The kth component of this vector, $1 \leq k < n$, is $\mathcal{MI}_n^k$, representing the bigram mutual information of $x_1 \dots x_k$ and $x_{k+1} \dots x_n$. The meaning of this vector function will be further explained in the next section. Finally, the generalized mutual information function of two adjacent elements xy in an n-gram is denoted by $\mathcal{GMI}_n(x,y)$.

Mutual Information

The bigram mutual information of two events is a measure of the interdependence of these events in sequence. In applying the concept of mutual information to the analysis of sentences, we are concerned with more than just the interdependence of a bigram. In order to take into account the context of the bigram, the interdependence of part-of-speech n-grams (sequences of n parts-of-speech) must be considered. Thus, we consider an n-gram as a bigram of an n_1-gram and an n_2-gram, where $n_1 + n_2 = n$. The mutual information of this bigram is

$$\mathcal{MI}(n_1\text{-gram}, n_2\text{-gram}) = \log \frac{\mathcal{P}[n\text{-gram}]}{\mathcal{P}[n_1\text{-gram}]\mathcal{P}[n_2\text{-gram}]}. \tag{3}$$

Notice that there are $(n-1)$ ways of partitioning an n-gram. Thus, for each n-gram, there is an $(n-1)$ vector of mutual information values. For a given n-gram $x_1 \dots x_n$, we can define the mutual information values of x by:

$$\begin{aligned}
\mathcal{MI}_n^k(x_1 \dots x_n) &= \mathcal{MI}(x_1 \dots x_k, x_{k+1} \dots x_n) && (4)\\
&= \log \frac{\mathcal{P}(x_1 \dots x_n)}{\mathcal{P}(x_1 \dots x_k)\mathcal{P}(x_{k+1} \dots x_n)} && (5)
\end{aligned}$$

where $1 \leq k < n$.

Notice that, in the above equation, for each $\mathcal{MI}_n^k(x)$, the numerator, $\mathcal{P}(x_1 \dots x_n)$, remains the same while the denominator, $\mathcal{P}(x_1 \dots x_k)\mathcal{P}(x_{k+1} \dots x_n)$, depends on k. Thus, the mutual information value achieves its minimum at the point where the denominator is maximized. The empirical claim to be tested in this paper is that the minimum is achieved when the two components of this n-gram are in two different constituents, i.e. when $x_k x_{k+1}$ is a distituent. Our experiments show that this claim is largely true with a few interesting exceptions.

The motivation for this claim comes from examining the characteristics of n-grams which contain pairs of constituents. Consider a tag sequence, $x_1 \dots x_n$, which is composed of two constituents $x_1 \dots x_k$ and $x_{k+1} \dots x_n$. Since $x_1 \dots x_k$ is a constituent, $x_1 \dots x_{k-1}$ is very likely to be followed by x_k. Thus,

$$\mathcal{P}(x_1 \dots x_k) \approx \mathcal{P}(x_1 \dots x_{k-1}). \tag{6}$$

By the same logic,

$$\mathcal{P}(x_{k+1} \dots x_n) \approx \mathcal{P}(x_{k+2} \dots x_n). \tag{7}$$

On the other hand, assuming x_k and x_{k+1} are uncorrelated (in the general case),

$$\mathcal{P}(x_k \dots x_n) \ll \mathcal{P}(x_{k+1} \dots x_n) \tag{8}$$

and

$$\mathcal{P}(x_1 \dots x_{k+1}) \ll \mathcal{P}(x_1 \dots x_k). \tag{9}$$

Therefore,

$$\begin{aligned}
&\mathcal{MI}(x_1 \dots x_k, x_{k+1} \dots x_n) \\
&= \log \frac{\mathcal{P}(x_1 \dots x_n)}{\mathcal{P}(x_1 \dots x_k)\mathcal{P}(x_{k+1} \dots x_n)} && (10)\\
&\approx \log \frac{\mathcal{P}(x_1 \dots x_n)}{\mathcal{P}(x_1 \dots x_{k-1})\mathcal{P}(x_{k+1} \dots x_n)} && (11)\\
&> \log \frac{\mathcal{P}(x_1 \dots x_n)}{\mathcal{P}(x_1 \dots x_{k-1})\mathcal{P}(x_k \dots x_n)} && (12)\\
&= \mathcal{MI}(x_1 \dots x_{k-1}, x_k \dots x_n). && (13)
\end{aligned}$$

By applying a symmetry argument and using induction, the above logic suggests the hypothesis that, in the general case, if a distituent exists in an n-gram, it should be found where the minimum value of the mutual information vector occurs.

There is no significance to the individual mutual information values of an n-gram other than the minimum; however, the distribution of the values is significant. If all the values are very close together, then, while the most likely location of the distituent is still where the minimum occurs, the confidence associated with this selection is low. Conversely, if these values are distributed over a large range, and the minimum is much lower than the maximum, then the confidence is much higher that there is a distituent where the minimum occurs. Thus, the standard deviation of the mutual information values of an n-gram is an estimate of the confidence of the selected distituent.

Generalized Mutual Information

Although bigram mutual information can be extended simply to n-space by the technique described in the previous section, this extension does not satisfy the needs of a distituent parser. A distituent parsing technique attempts to select the most likely distituents based on its statistic. Thus, a straightforward approach would assign each potential distituent a single real number corresponding to the extent to which its context suggests it is a distituent. But the simple extension of bigram mutual information assigns each potential distituent a number for each n-gram of which it is a part. The question remains how to combine these numbers in order to achieve a valid measure of distituency.

Our investigations revealed that a useful way to combine mutual information values is, for each possible distituent xy, to take a weighted sum of the mutual information values of all possible pairings of n-grams ending with x and n-grams beginning with y, within a fixed size window. So, for a window of size $w = 4$, given the context $x_1x_2x_3x_4$, the generalized mutual information of x_2x_3 :

$$\mathcal{GMI}_4(x_1x_2, x_3x_4),$$

$$= k_1\mathcal{MI}(x_2, x_3) + k_2\mathcal{MI}(x_2, x_3x_4) + \tag{14}$$

$$k_3\mathcal{MI}(x_1x_2, x_3) + k_4\mathcal{MI}(x_1x_2, x_3x_4) \tag{15}$$

which is equivalent to

$$\log\left(k\frac{\mathcal{P}[x_2x_3]\mathcal{P}[x_2x_3x_4]\mathcal{P}[x_1x_2x_3]\mathcal{P}[x_1x_2x_3x_4]}{[\mathcal{P}[x_2]\mathcal{P}[x_3]\mathcal{P}[x_1x_2]\mathcal{P}[x_3x_4]]^2}\right) \tag{16}$$

In general, the generalized mutual information of any given bigram xy in the context $x_1 \ldots x_{i-1}xyy_1 \ldots y_{j-1}$ is equivalent to

$$\log\left(\frac{\prod_{X \text{crosses } xy} k_X\mathcal{P}[X]}{\prod_{X \text{does not cross } xy} \mathcal{P}[X]^{(i+j)/2}}\right). \tag{17}$$

This formula behaves in a manner consistent with one's expectation of a generalized mutual information statistic. It incorporates all of the mutual information data within the given window in a symmetric manner. Since it is the sum of bigram mutual information values, its behavior parallels that of bigram mutual information.

The weighting function which should be used for each term in the equation was alluded to earlier. The standard deviation of the values of the bigram mutual information vector of an n-gram is a valid measure of the confidence of these values. Since distituency is indicated by mutual information minima, the weighting function should be the reciprocal of the standard deviation.

In summary, the generalized mutual information statistic is defined to be:

$$\mathcal{GMI}_{(i+j)}(x_1 \ldots x_i, y_1 \ldots y_j)$$

$$= \sum_{\substack{X \text{ ends with } x_i \\ Y \text{ begins with } y_1}} \frac{1}{\sigma_{XY}}\mathcal{MI}(X, Y), \tag{18}$$

where σ_{XY} is the standard deviation of the $\mathcal{MI}^k_{|XY|}$ values within XY.

The Parsing Algorithm

The generalized mutual information statistic is the most theoretically significant aspect of the mutual information parser. However, if it were used in a completely straightforward way, it would perform rather poorly on sentences which exceed the size of the maximum word window. Generalized mutual information is a local measure which can only be compared in a meaningful way with other values which are less than a word window away. In fact, the further apart two potential distituents are, the less meaningful the comparison between their corresponding $\mathcal{GMI}$ values. Thus, it is necessary to compensate for the local nature of this measure algorithmically.

```
He directed the cortege of autos to the dunes
near Santa Monica.
```

Figure 1: Sample sentence from the Brown Corpus

We will describe the parsing algorithm as it parses a sample sentence (Figure 1) selected from the section of the Tagged Brown Corpus which was *not* used for training the parser. The sample sentence is viewed by the parser as a tag sequence, since the words in the sentence are not accounted for in the parser's statistical model. The sentence is padded on both sides with $w - 1$ blank tags (where w is the maximum word window size) so there will be adequate context to calculate generalized mutual information values for all possible distituents in the sentence.

A bigram mutual information value vector and its standard deviation are calculated for each n-gram in the sentence, where $2 \leq n \leq 10$.[2] If the frequency of an n-gram is below a certain threshold (< 10, determined experimentally), then the mutual information values are all assumed to be 1, indicating that no information is given by that n-gram. These values are calculated once for each sentence and referenced frequently in the parse process.

Distituent	Pass 1	DG	Pass 2	Pass 3
pro verb	3.28	*3.28*	*3.28*	3.28
verb det	3.13	*3.13*	*3.13*	3.13
det noun	11.18	11.18		
noun prep	11.14	$-\infty$	8.18	
prep noun	1.20	1.20		
noun prep	7.41	$-\infty$	*3.91*	*2.45*
prep det	16.89	16.89	10.83	
det noun	16.43	*16.43*		
noun prep	12.73	$-\infty$	*7.64*	4.13
prep noun	7.36	7.36		

Figure 2: Parse node table for sample sentence

Next, a parse node is allocated for each tag in the sentence. A generalized mutual information value is computed for each possible distituent, i.e. each pair of parse nodes, using the previously calculated bigram mutual information values. The resulting parse node table for the sample sentence is indicated by Pass 1 in the parse node table (Figure 2).

At this point, the algorithm deviates from what one might expect. As a preprocessing step, the distituent grammar is invoked to flag any known distituents by replacing their $\mathcal{GMI}$ value with $-\infty$. The results of this phase are indicated in the DG column in the parse node table.

The first w tags in the sentence are processed using an n-ary-branching recursive function which branches at the minimum $\mathcal{GMI}$ value of the given window. The local minima at which branching occurs in each pass of the parse are indicated by italics in the parse node table. One should note that marginal differences between $\mathcal{GMI}$ values are not considered significant. So, for instance, the distituency of *pro verb* (3.28) is considered equivalent to the distituency of *verb det* (3.13) in the sample sentence. This behavior results in n-ary trees instead of binary trees.

[2]The optimal maximum word window size, $w = 10$, was determined experimentally. However, since there were only 46 11-grams and 15 12-grams which occurred more than 10 times in the training corpus, it is obvious why virtually no information is gained by expanding this window beyond 10. By training the parser on a larger corpus, or a corpus with a higher average sentence length, the optimal maximum word window size might be larger.

Instead of using this tree in its entirety, only the nodes in the leftmost constituent leaf are pruned. The rest of the nodes in the window are thrown back into the pool of nodes. The same process is applied to the last w remaining tags in the sentence, but this time the rightmost constituent leaf is pruned from the resulting parse tree. The algorithm is applied again to the leftmost w remaining tags, and then the rightmost w tags, until no more tags remain. The first pass of the parser is complete, and the sentence has been partitioned into constituents (Figure 3).

```
(He) (directed) (the cortege) (of autos)
(to) (the dunes) (near Santa Monica)
```

Figure 3: Constituent structure after Pass 1

In pass 2, a parse node is assigned to each constituent unit determined from the first pass, GMI values are calculated for these parse nodes, and the left-right pruning algorithm is applied to them.

The algorithm terminates when no new structure has been ascertained on a pass, or when the lengths of two adjacent constituents sum to greater than w. In both cases, the parser can extract no more information about the distituency of the nodes from the statistics available. In the first case, the resulting distituency confidence values are too close together to determine distituency; and in the second case, since the word window can no longer span a potential distituent, the algorithm must give up. After the third pass of the algorithm, the sample sentence is partitioned into two adjacent constituents, and thus the algorithm terminates, with the result in figure 4.

```
(He (directed ((the cortege) (of autos)))
    ((to (the dunes))
     (near Santa Monica)))
```

Figure 4: Resulting constituent structure after Pass 3

Processing only a word-window of information at a time and pruning the leftmost and rightmost leaves of the resulting subtrees are the keys to minimizing the error introduced by the use of a non-global, estimated statistic. Since we know that the parser tends to make errors, our goal is to minimize these errors. Finding constituents in the middle of a sentence requires locating two distituents, whereas finding them at the beginning or end of a sentence requires locating only one distituent. Thus, pruning constituents from the beginning and end of a tag sequence produces a far more accurate partitioning of the sentence than trying to guess them all at once.

It is important to note that, on a given pass of the parser, many of the 'constituents' which are pruned are actually only single nodes. For instance, in the sample sentence, the first pass partitions the phrase "to the dunes" as "(to) (the dunes)." A subsequent pass of the parsing algorithm attaches the preposition to the noun phrase (although the parser has no knowledge of these constituent names). However, once the entire phrase is found to be a constituent, it is not scanned for any further structural information. Thus, if the first pass had grouped the phrase as "(to the dunes)," then the noun phrase within the prepositional phrase would never be marked as a constituent.

As a result of this behavior, the prepositional phrase "near Santa Monica" will not attach to the noun phrase "the dunes" (or to the noun "dunes" as many linguists believe it should) once the prepositional phrase is formed. Therefore, the parser output for the sample sentence has one error.

Results

Evaluating the accuracy of a natural language parser is as difficult as writing a full grammar for that natural language, since one must make decisions about grammar rules in order to decide what is an error and what is not. Serious thought must be put into questions like: where does a conjunction bind in a conjunct, and does it matter? or where do prepositional phrases attach, and can we even decide? These very problems are the reason we sought an alternative to a grammar-based parser. Thus, while the error rate for short sentences (15 words or less) with simple constructs can be determined very accurately, the error rate for longer sentences is more of an approximation than a rigorous value.

Our parser is very good at parsing short sentences of unrestricted text without conjunctions. On these sentences, the parser averages close to one error per sentence. However, if free text with conjunctions is included, the performance falls to close to two errors per sentence. An error is defined as a misparse which can be corrected by moving one subtree.

As one would expect, our parser's performance is not as accurate for longer sentences, but it is certainly respectable. On sentences between 16 and 30 tokens in length, the parser averages between 5 and 6 errors per sentence. However, in nearly all of these longer sentences and many of shorter ones, at least one of the errors is caused by confusion about conjuncts, especially sentences joined by conjunctions. Considering the parser is trained on n-grams with a word window no larger than 10 tokens, it is not surprising that it fails on sentences more than twice that size. Given a larger training corpus with a significant number of these long sentences, the maximum word window could be increased and the parser would undoubtedly improve on these longer sentences.

The output from the mutual information parser is unique in that it gives both more and less information than most other statistical parsers. Most statistical parsers depend on internal grammar rules which allow them both to estimate and to label sentence structure. Once again, because of the complexity of natural language grammars, these approaches can only extract limited levels of structure. Hindle's FIDDITCH parser (1988) attempts to extract not only sentence structure but also noun classifications using cooccurrence of word pairs, another variation of bigram mutual information. While his technique performs the noun classification task extremely well, it does not seriously attempt to parse sentences completely, since its grammar cannot handle complex sentence structures. Our parser is capable of determining all levels of sentence structure, although it is incapable of labeling the resulting constituents.

Conclusion

The performance of this parsing algorithm demonstrates that a purely syntactic, stochastic technique can effectively determine all levels of sentence structure with a relatively high degree of accuracy. The most important question to ask at this juncture is: where do we go from here?

An immediate extension of this research would be to apply a simple grammar-based filter to each pass of our statistical parser to verify the validity of the constituents it determines. Applying a very simple grammar which defines only constituency of terminal symbols would eliminate many of the errors made by our parser.

The implementation of an effective deterministic parsing algorithm, however, should not overshadow the real discovery of this research. The generalized mutual information statistic is a powerful statistical measure which has many other applications in natural language processing. Bigram mutual information has been applied to many different problems requiring n-gram analysis. It would be interesting to reinvestigate these problems using generalized mutual information. In particular, Hindle's noun classification work (Hindle 1988) and Church's part-of-speech assignment (Church 1988) might benefit from this statistic.

Another way in which this research might be used is as a supplement to a grammar-based parser. The distituent parsing method could be used in whole as a pre-processor to supply hints for a grammar-based parser; or it could be used incrementally in a bottom-up parsing technique to provide guidelines for search so that non-deterministic algorithms do not realize their worst-case inefficiency.

Another interesting possibility is to use the generalized mutual information statistic to extract a grammar from a corpus. Since the statistic is consistent, and its window can span more than two constituents, it could be used to find constituent units which occur with the same distribution in similar contexts.

There are many problems in natural language processing which cannot be solved *efficiently* by grammar-based algorithms and other problems which cannot be solved *accurately* by stochastic algorithms. This research suggests that the solution to some of these problems is a combination of both.

References

[1] Church, K. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of the Second Conference on Applied Natural Language Processing. Austin, Texas.

[2] Church, K.; and Gale, W. 1990. Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of English Bigrams. *Computers, Speech and Language.*

[3] Church, K.; and Hanks, P. 1989. Word Association Norms, Mutual Information, and Lexicography. In Proceedings of the 27th Annual Conference of the Association of Computational Linguistics.

[4] Fano, R. 1961. *Transmission of Information.* New York, New York: MIT Press.

[5] Francis, W.; and Kucera, H. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar.* Boston, Mass.: Houghton Mifflin Company.

[6] Hindle, D. 1988. Acquiring a Noun Classification from Predicate-Argument Structures. Bell Laboratories.

[7] Jelinek, F. 1985. Self-organizing Language Modeling for Speech Recognition. IBM Report.

[8] Katz, S. M. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 3.*