

Deducing Linguistic Structure from the Statistics of Large Corpora

Eric Brill, David Magerman, Mitchell Marcus, Beatrice Santorini

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

1 Introduction

Within the last two years, approaches using both stochastic and symbolic techniques have proved adequate to deduce lexical ambiguity resolution rules with less than 3-4% error rate, when trained on moderate sized (500K word) corpora of English text (e.g. Church, 1988; Hindle, 1989). The success of these techniques suggests that much of the grammatical structure of language may be derived automatically through distributional analysis, an approach attempted and abandoned in the 1950s.

We describe here two experiments to see how far purely distributional techniques can be pushed to automatically provide both a set of part of speech tags for English, and a grammatical analysis of free English text. We also discuss the state of a tagged NL corpus to aid such research (now amounting to 4 million words of hand-corrected part-of-speech tagging).

In the experiment described in Section 2, we have developed a constituent boundary parsing algorithm which derives an (unlabelled) bracketing given text annotated for part of speech as input. This method is based on the hypothesis that constituent boundaries can be extracted from a given part-of-speech n -gram by analyzing the mutual information values within the n -gram, extended to a new generalization of the information theoretic measure of *mutual information*. This hypothesis is supported by the performance of an implementation of this parsing algorithm which determines recursively nested sentence structure, with an error rate of roughly 2 misplaced boundaries for test sentences of length 10-15 words, and five misplaced boundaries for sentences of 15-30 tokens. To combat a limited set of specific circumstances in which the hypothesis fails, we use a small (4 rule, 8 symbol) *distituent grammar*, which indicates when two parts of speech *cannot* remain in the same constituent.

In another experiment, described in Section 3, we investigate whether a distributional analysis can discover

a part of speech tag set which might prove adequate to support experiments like that discussed above. We have developed a similarity measure which accurately clusters closed-class lexical items of the same grammatical category, excepting words which are ambiguous between multiple parts of speech.

2 A Mutual Information Parser

2.1 Introduction

In this section, we 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. We view part-of-speech sequences as stochastic events and apply 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 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.

This hypothesis assumes that, given any constituent n -gram, $a_1 a_2 \dots a_n$, the probability of that constituent occurring is usually significantly higher than the probability of $a_1 a_2 \dots a_n a_{n+1}$ 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. As it turns out, however, there are cases in which this assumption is not valid, but only a handful of these cases are responsible for a majority of the errors made by the parser. To deal with these cases, our algorithm includes what we will call a distituent grammar — a list of tag pairs which *cannot* be adjacent within a constituent. One such pair is *noun prep*, since 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

¹This 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. Thanks to Ken Church, Stuart Shieber, Max Mintz, Aravind Joshi, Lila Gleitman and Tom Veatch for their valued suggestions and discussion.

a unit. Our current constituent grammar consists of four rules of two tokens each.

Our current implementation of this parsing algorithm determines a recursive unlabeled bracketing of unrestricted English text. The generalized mutual information statistic and the constituent 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 2). 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 reasonably efficient, parsing a 35,000 word corpus in under 10 minutes on a Sun 4/280.

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.² 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.

2.2 Mutual Information Statistics

The mutual information statistic (Fano 1961) is a measure of the interdependence of two signals in a message. It is a function of the probabilities of the two events:

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

In this paper, the events x and y will be part-of-speech *n*-grams (instead of single parts-of-speech, as in some earlier work).

Experiments that we will not report here show that simple mutual information statistics computed on *n*-gram sequences are not sufficient for the task at hand. Instead, we have moved to a statistic which we will call "generalized mutual information," because it is a generalization of the mutual information of part-of-speech

²The corpus we use to train our parser is the Tagged Brown Corpus (Francis and Kučera, 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.

bigrams into *n*-space. Generalized mutual information uses the context on both sides of adjacent parts-of-speech to determine a measure of its constituency in a given sentence.

While our constituent 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.

2.2.1 Generalized Mutual Information

In applying the concept of mutual information to the analysis of sentences, 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

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

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:

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

where $1 \leq k < n$.

Notice that, in the above equation, for each $MI_n^k(x)$, 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 constituent. Our experiments show that this claim is largely true with a few interesting exceptions.

A straightforward approach would assign each potential constituent a single real number corresponding to the extent to which its context suggests it is a constituent. But the simple extension of bigram mutual information assigns each potential constituent 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 constituency.

Our investigations revealed that a useful way to combine mutual information values is, for each possible constituent 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 $x_1 x_2 x_3 x_4$, the generalized mutual information of $x_2 x_3$:

$$\begin{aligned} GMI_4(x_1 x_2, x_3 x_4) &= k_1 MI(x_2, x_3) + k_2 MI(x_2, x_3 x_4) + \\ &= k_3 MI(x_1 x_2, x_3) + k_4 MI(x_1 x_2, x_3 x_4) \end{aligned}$$

which is equivalent to

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

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

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

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 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 distitency is indicated by mutual information minima, we use the reciprocal of the standard deviation as a weighting function.

2.3 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 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

In Magerman and Marcus (1990) we describe the parsing algorithm in detail, and trace the parsing of 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.

A bigram mutual information value vector and its standard deviation are calculated for each n -gram in the sentence, where $2 \leq n \leq 10$. 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	<i>3.28</i>	<i>3.28</i>	3.28
verb det	3.13	<i>3.13</i>	<i>3.13</i>	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	$-\infty$	<i>3.91</i>	<i>2.45</i>
prep det	16.89	16.89	10.83	
det noun	16.43	<i>16.43</i>		
noun prep	12.73	$-\infty$	<i>7.64</i>	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 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 GMI value of the given window, with marginal differences between GMI values ignored. The local minima at which branching occurs in each pass of the parse are indicated by italics in the parse node table.

Instead of using this tree in its entirety, only the nodes in the leftmost and rightmost constituent leaves are pruned. The rest of the nodes in the window are thrown back into the pool of nodes. The algorithm is applied again to the leftmost and rightmost w remaining 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

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 . After two more passes of the algorithm, the sample sentence is partitioned into two adjacent constituents, and thus the algorithm terminates, with the result in figure 4. In this example, the prepositional phrase “near Santa Monica” is not attached to the noun phrase “the dunes” as it

should be; therefore, the parser output for the sample sentence has one error.

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

Figure 4: Resulting constituent structure after Pass 3

2.4 Results

A careful evaluation of this parser, like any other, requires some “gold standard” against which to judge its output. Soon, we will be able to use the skeletal parsing of the Penn Treebank we are about to begin producing to evaluate this work (although evaluating this parser against materials which we ourselves provide is admittedly problematic). For the moment, we have simply graded the output of the parser by hand ourselves. While the error rate for short sentences (15 words or less) with simple constructs is accurate, the error rate for longer sentences is more of an approximation than a rigorous value.

On unconstrained free text from a reserved test corpus, the parser averages about two errors per sentence for sentences under 15 words in length. On sentences between 16 and 30 tokens in length, it averages between 5 and 6 errors per sentence. In nearly all of these longer sentences and many of shorter ones, at least one of the errors is caused by confusion about conjuncts.

One 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. Given the results of the next section, it may well be possible to use automatic techniques to first determine a first approximation to the set of word classes of a language, given only a large corpus of text, and then extract a grammar for that set of word classes. Such a goal is very difficult, of course, but we believe that it is worth pursuing. In the end, we believe that this, like many problems in natural language processing, cannot be solved *efficiently* by grammar-based algorithms nor *accurately* by purely stochastic algorithms. We believe strongly that the solution to some of these problems may well be a combination of both approaches.

3 Discovering the Word Classes of a Language

3.1 Introduction

As we ask immediately above, to what extent is it possible to discover by some kind of distributional analysis the kind of part-of-speech tags upon which our mutual information parser depends? In this section, we examine the possibility of using distributional analysis to dis-

cover the feature set and word classes of a language.³ It is based upon the following idea, a variant of the distributional analysis methods from Structural Linguistics (Harris 51, Harris 68): features license the distributional behavior of lexical items. At the two extremes, a word with no features would not be licensed to appear in any context at all, whereas a word marked with all features of the language would be licensed to appear in every possible context.

3.2 The Algorithm

The feature discovery system works as follows. First, a large amount of text is examined to discover the frequency of occurrence of different bigrams.⁴ Based upon this data, the system groups words into classes. Two words are in the same class if they can occur in the same contexts. In order to determine whether x and y belong to the same class, the system first examines all bigrams containing x . If for a high percentage of these bigrams, the corresponding bigram with y substituted for x exists in the corpus, then it is likely that y has all of the features that x has (and maybe more). If upon examining the bigrams containing y the system is able to conclude that x also has all of the features that y has, it then concludes that x and y are in the same class.

For every pair of bigrams, the system must determine how much to weigh the presence of those bigrams as evidence that two words have features in common. For instance, assume: (a) the bigram *the boy* appears many times in the corpus being analyzed, while *the sits* never occurs. Also assume: (b) the bigram *boy the* (as in *the boy the girl kissed . . .*) occurs once and *sits the* never occurs. Case (a) should be much stronger evidence that *boy* and *sits* are not in the same class than case (b). For each bigram ax occurring in the corpus, evidence offered by the presence (or absence) of the bigram ay is scaled by the frequency of ax in the text divided by the total number of bigrams containing x on their right hand side. Since the end-of-phrase position is less restrictive, we would expect each bigram involving this position and the word to the right of it to occur less frequently than bigrams of two phrase-internal words. By weighing the evidence, bigrams which cross boundaries will be weighed less than those which do not.

3.2.1 The Specifics

The function $\text{implies}(x,y)$ calculates the likelihood (on a scale of [0..1]) that word y contains all of the features of word x . For example, we would expect the value of $\text{implies}('a', 'the')$ to be close to 1, since 'the' can occur in any context which 'a' can occur in. Note that: $\text{implies}(x,y) \wedge \text{implies}(y,x)$ iff x and y are in the same class.

³We consider the set of features of a particular language to be all attributes which that language makes reference to in its syntax.

⁴For this experiment, we take a very local view of context, only considering bigrams.

$$\text{implies}(x, y) = \frac{\text{leftimply}(x, y) + \text{rightimply}(x, y)}{2}$$

The function **leftimply**(*x*,*y*) is the likelihood (on a scale of [0..1]) that *y* contains all of the features of *x*, where this likelihood is derived from looking at bigrams of the form: *x* α . **rightimply**(*x*,*y*) derives the likelihood by examining all bigrams of the form: α *x*. **bothoccur**(α , β) is 1 if both bigrams α and β occur in the corpus, and β occurs with a frequency at least 1/THRESHOLD of that of α , for some THRESHOLD.⁵ **bothoccur** accounts for the fact that we cannot expect the distribution of two equivalent words over bigrams to be precisely the same, but we would not expect the two distributions to be too dissimilar either.

$$\text{leftimply}(x, y) = \frac{\sum_{z \in \Omega} (\text{percentage}_{\text{left}}(x, z) * \text{bothoccur}_{\text{left}}(zx, yz))}{\sum_{z \in \Omega} \text{percentage}_{\text{left}}(x, z)}$$

$$\text{rightimply}(x, y) = \frac{\sum_{z \in \Omega} (\text{percentage}_{\text{right}}(x, z) * \text{bothoccur}_{\text{right}}(zx, zy))}{\sum_{z \in \Omega} \text{percentage}_{\text{right}}(z, x)}$$

$$\text{bothoccur}_{\text{left}}(ab, cd) = \begin{cases} 1 & \text{if bigrams } ab \text{ and } cd \text{ appear in the corpus and} \\ & \text{percentage}_{\text{left}}(c,d) \geq (1/\text{THRESHOLD} * \\ & \text{percentage}_{\text{left}}(a,b)) \\ 0 & \text{otherwise} \end{cases}$$

When computing the relation between *x* and all other words, we use the following function, **percentage**, to weigh the evidence (as described above), where **count**(*ab*) is the number of occurrences of the bigram *ab* in the corpus, and **numright**(*x*) (**numleft**(*x*)) is the total number of bigrams with *x* on their right hand side (left hand side).

$$\text{percentage}_{\text{left}}(x, y) = \frac{\text{count}(xy)}{\text{numleft}(x)}$$

$$\text{percentage}_{\text{right}}(x, y) = \frac{\text{count}(yx)}{\text{numright}(x)}$$

For all pairs of words, *x* and *y*, we calculate **implies**(*x*,*y*) and **implies**(*y*,*x*). We can then find word classes in the following way. We first determine a threshold value, where a stronger value will result in more specific classes. Then, for each word *x*, we find all words

⁵In the experiments we ran, we found THRESHOLD = 6 to give the best results. This value was found by examining the values of implication found between *the*, *a* and *an*.

y such that both **implies**(*x*,*y*) and **implies**(*y*,*x*) are greater than the threshold. We next take the transitive closure of pairs of sets with nonempty intersection over all of these sets, and the result is a set of sets, where each set is a word class. Classes of different degrees of specificity are found by varying the degree of similarity between distributions needed to conclude that two words are in the same class. If a high degree of similarity is required, all words in a class will have the same features. If a lower degree of similarity is required, then words in a class must have most, but not all, of the same features.

3.3 The Experiment

To test the algorithm discussed above, we ran the following experiment. First, the number of occurrences of each bigram in the corpus was determined. Statistics on distribution were determined by examining the complete Brown Corpus (Francis 82), where infrequently occurring open-class words were replaced with their part-of-speech tag. We then ran the program on a group of words including all closed-class words which occurred more than 250 times in the corpus, and the most frequently occurring open-class words. Note that the system attempted to determine the relations between these words; this does not mean that it only considered bigrams $\alpha\beta$, where both α and β were from this list of words which were being partitioned. All bigrams which occurred more than 5 times were considered in the distributional analysis.

3.4 Analysis of the Experiment

The program successfully partitioned words into word classes.⁶ In addition, it was able to find more fine-grained features. Among the features found were: [possessive-pronoun], [singular-determiner], [definite-determiner], [wh-adjunct] and [pronoun+be]. A description of some of the word classes the program discovered can be found in Appendix A.

3.5 The Psychological Plausibility of Distributional Analysis

If a child does not know a priori what features are used in her language, there are two ways in which she can acquire this information: by using either syntactic or semantic cues. The child could use syntactic cues such as the method of distributional analysis described in this paper. The child might also rely upon semantic cues. There is evidence that children use syntactic rather than semantic cues in classifying words. Peter Gordon (Gordon 85) ran an experiment where the child was presented with an object which was given a made up name. For objects with semantic properties of count nouns (mass nouns), the word was used in lexical environments which only mass nouns (count nouns) are permitted to be in. Gordon showed that the children overwhelmingly used

⁶One exception was the class of pronouns. Since [+nominative] and [-nominative] pronouns do not have similar distribution, they were not found to be in the same class.

	Raw no. of words	Times tagged	Total no. of words
Brown Corpus	1,159,381	1	1,159,381
Library of America	159,267	2	318,534
DOE abstracts	199,928	2	399,856
Dow Jones Corpus	2,644,618	1	2,644,618
Grand total	4,163,194		4,522,389

Table 1: Number of words tagged

the distributional cues and not the semantic cues in classifying the words. Virginia Gathercole (Gathercole 85) found that “children do not approach the co-occurrence conditions of *much* and *many* with various nouns from a semantic point of view, but rather from a morphosyntactic or surface-distributional one.” Yonata Levy (Levy 83) examined the mistakes young children make in classifying words. The mistakes made were not those one would expect the child to make if she were using semantic cues to classify words.

4 Penn Treebank

In this section, we report some recent performance measures of the Penn Treebank Project.

To date, we have tagged over 4 million words by part of speech (cf. Table 1). We are tagging this material with a much simpler tagset than used by previous projects, as discussed at the Oct. 1989 DARPA Workshop. The material is first processed using Ken Church’s tagger (Church 1988), which labels it as if it were Brown Corpus material, and then is mapped to our tagset by a SED-script. Because of fundamental differences in tagging strategy between the Penn Treebank Project and the Brown project, the resulting mapping is about 9% inaccurate, given the tagging guidelines of the Penn Treebank project (as given in 40 pages of explicit tagging guidelines). This material is then hand-corrected by our annotators; the result is consistent within annotators to about 3% (cf. Table 3), and correct (again, given our tagging guidelines) to about 2.5% (cf. Table 2), as will be discussed below. We intend to use this material to retrain Church’s tagger, which we then believe will be accurate to less than 3% error rate. We will then adjudicate between the output of this new tagger, run on the same corpus, and the previously tagged material. We believe that this will yield well below 1% error, at an additional cost of between 5 and 10 minutes per 1000 words of material. To provide exceptionally accurate bigram frequency evidence for retraining the automatic tagger we are using, two subcorpora (Library of America, DOE abstracts) were tagged twice by different annotators, and the Library of America texts were adjudicated by a third annotator, yielding ~160,000 words tagged with an accuracy estimated to exceed 99.5%.

Table 2 provides an estimate of error rate for part-of-speech annotation based on the tagging of the sample described above. Error rate is measured in terms of the

Tagger	No. of errors	Error rate
RF	105	1.9
CH	151	2.8
MAM	127	2.3
MP	158	2.9
MW	136	2.5
Mean	135	2.5

Table 2: Error rates

	CH	MAM	MP	MW
RF	2.6%	3.5%	3.2%	3.0%
CH	-	2.9%	3.9%	3.7%
MAM	-	-	3.3%	2.7%
MP	-	-	-	2.8%
Mean:				3.2%

Table 3: Inter-annotator inconsistency

number of disagreements with a benchmark version of the sample prepared by Beatrice Santorini. We have also estimated the rate of inter-annotator inconsistency based on the tagging of the sample described above (cf. Table 3). Inconsistency is measured in terms of the proportion of disagreements of each of the annotators with each other over the total number of words in the test corpus (5,425 words).

Table 4 provides an estimate of speed of part-of-speech annotation for a set of ten randomly selected texts from the Dow Jones Corpus (containing a total of 5,425 words), corrected by each of our annotators. The annotators were thoroughly familiar with the genre, having spent over three months immediately prior to the experiment correcting texts from the same Corpus. Given that the average productivity overall of our project has been between 3,000-3,500 words per hour of time billed by our annotators, it appears that our strategy of hiring annotators for no more than 3 hours a day has proven to be quite successful.

Finally, the summary statistics in Table 5 provide an estimate of improvement of annotation speed as a function of familiarity with genre. We compared the annotators’ speed on two samples of the Brown Corpus (10 texts) and the Dow Jones Corpus (100 texts). We examined the first and last samples of each genre that the

Tagger	Time (in minutes)	Words per hour	Minutes per 1,000 words
RF	68	4,804	12.5
CH	79	4,129	14.5
MAM	57	5,751	10.4
MP	74	4,423	13.3
MW	100	3,268	18.3
Mean	76	4,283	14.0

Table 4: Speed of part-of-speech annotation

		Words per hour	Minutes per 1,000 words
Early	Brown	2,816	21.3
	Dow Jones	1,711	35.1
	Mean	2,621	22.9
Late	Brown	3,483	17.2
	Dow Jones	3,641	16.5
	Mean	3,511	17.1
Improvement		34%	25%

Table 5: Speed as function of familiarity with genre

annotators tagged; in each case, more than two months of experience lay between the samples.

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 Kučera, H. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston, Mass.: Houghton Mifflin Company.
- [6] Gathercole, V. 'He has too much hard questions': the acquisition of the linguistic mass-count distinction in *much* and *many*. *Journal of Child Language*, 12: 395-415.
- [7] Gordon, P. Evaluating the semantic categories hypothesis: the case of the count/mass distinction. *Cognition*, 20: 209-242.
- [8] Harris, Z.S. (1951) *Structural Linguistics*. Chicago: University of Chicago Press.
- [9] Harris, Z.S. (1968) *Mathematical Structures of Language*. New York: Wiley.
- [10] Hindle, D. 1988. Acquiring a Noun Classification from Predicate-Argument Structures. Bell Laboratories.
- [11] Jelinek, F. 1985. Self-organizing Language Modeling for Speech Recognition. IBM Report.
- [12] 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.
- [13] Levy, Y. It's frogs all the way down. *Cognition*, 15: 75-93.
- [14] Magerman, D. and Marcus, M. Parsing a Natural Language Using Mutual Information Statistics, *Proceedings of AAAI-90*, Boston, Mass (forthcoming).
- [15] Pinker, S. *Learnability and Cognition*. Cambridge: MIT Press.

