Current Practice in Part of Speech Tagging and Suggestions for the Future

Kenneth Ward Church
Bell Laboratories
600 Mountain Ave.
Murray Hill, N.J., USA
908-582-5325
kwc@research.att.com

Abstract

As a result of corpus collection efforts such as the Tagged Brown Corpus, the Penn Treebank and similar efforts within the ICAME community, there are now quite a number of extremely successful part of speech tagging programs which make use of probabilities derived from corpus data. These programs work on unrestricted texts, with reasonable accuracy and efficiency. Most researchers report performance in the range of 95-99% ''correct'' by word, which is generally regarded by the computational linguistic community as a major advance over the level of performance offered by previously accepted technologies such as augmented transition networks (ATNs).

Much of this progress has been achieved because lexical probabilities (probability of a part of speech given the word) are now being estimated directly from corpus data, and can therefore be estimated much more accurately than before. Until recently, it had been common practice for most researchers in computational linguistics to concentrate their energies on modeling contextual constraints (e.g., grammar), which appears to be much less important than lexical probabilities (e.g., the dictionary), at least for the part of speech tagging application.

In my opinion, future work should continue to concentrate on lexical probabilities, which are actually much harder to estimate than one might have thought. The lexical probabilities tend to have more parameters than the contextual probabilities. Moreover, the relationships among lexical items turn out to be extremely subtle. One would hope that it would be possible to predict the lexical probabilities by making use of what we know about morphologically related forms. But even this apparently benign step is fraught with peril, we as will see.

1. Introduction

1.1 Problem Statement

It is well-known that part of speech depends on context. The word ‘‘table,’’ for example, can be a verb in some contexts (e.g., ‘‘He will table the motion’’) and a noun in others (e.g., ‘‘The table is ready’’). Quite a number of programs have recently been written which attempt to tag each word in an input text with the most likely part of speech. For example, the following output is produced by the program described in (Church, 1988) for the two ‘‘table’’ sentences just mentioned:

- He/PPS will/MD table/VB the/AT motion/NN ./.
- The/AT table/NN is/BEZ ready/JJ ./.

The notation is borrowed from Francis and Kucera (1982), pp. 6-8. PPS = subject pronoun; MD = modal; VB = verb (no inflection); AT = article; NN = noun; BEZ = present 3rd sg form of ‘‘to be’’; JJ = adjective. A more realistic example is presented in the appendix.

1.2 Motivation

Part of speech tagging is an important practical problem with potential applications in many areas including speech synthesis, speech recognition, spelling correction, proof-reading, query answering, machine translation and information retrieval. The author has been particularly interested in speech synthesis applications, where it is clear that pronunciation sometimes depends on part of speech. Consider the following three examples where pronunciation depends on part of speech. First, there are words like ‘‘wind’’ where the noun has a different vowel than the verb. That is, the noun ‘‘wind’’ has a short vowel as in ‘‘the wind is strong,’’ whereas the verb ‘‘wind’’ has a long vowel as in ‘‘Don’t forget to wind your watch.’’ Secondly, the pronoun ‘‘that’’ is stressed as in ‘‘Did you see THAT?’’ unlike the complementizer ‘‘that,’’ as in ‘‘It is a shame that [schwa] he’s leaving.’’ Thirdly, note the difference between ‘‘oily FLUID’’ and ‘‘TRANSMISSION fluid’’; as a general rule, an adjective-noun sequence such as ‘‘oily FLUID’’ is typically stressed on the right whereas a noun-noun sequence such as ‘‘TRANSMISSION fluid’’ is typically stressed on the left. These are but three of the many constructions which would sound more natural if the synthesizer had access to accurate part of speech information.

Perhaps the most important application of tagging programs is as a tool for future research. A number of large projects such as the Cobuild Dictionary (Sinclair *et al*, 1987) have recently been collecting large corpora (10-1000 million words) in order to better describe how language is actually used in practice:

> ‘‘For the first time, a dictionary has been compiled by the thorough examination of representative group of English texts, spoken and written, running to many millions of words. This means that in addition to all the tools of the conventional dictionary makers... the dictionary is based on hard, measurable evidence.’’ (Sinclair *et al*, 1987, p. xv)

It is likely that there will be more and more research projects collecting larger and larger corpora. A reliable parts program might greatly enhance the value of these corpora to many of these researchers.

1.3 The Statistical Approach

These days, most part of speech programs (e.g., Leech *et al.* (1983), Jelinek (1985), Deroualt and Merialdo (1986), Church (1988), DeRose (1988), Kupiec (1989), Ayuso *et al.* (1990), de Marcken (1990), Boggess *et al.* (1991), Merialdo (1991)) make use of the n-gram approximation, although there are a few recent exceptions (e.g., Heidorn *et al.* (1982), Martin *et al.* (1987), Hindle (1989), Karlsson (1990)). These

programs, which are based on probabilities derived from corpus data, are extremely successful; they work on unrestricted texts, with reasonable accuracy and efficiency.

Most of these statistical programs use a linear time dynamic programming algorithm to find an assignment of parts of speech to words that optimizes the product of

1. lexical probabilities (probability of observing part of speech i given word j), and
2. contextual probabilities (probability of observing part of speech *i* given *k* previous parts of speech).

I have always preferred to estimate the probabilities by training on a tagged corpus such as the Tagged Brown Corpus (Francis and Kucera, 1982), the Penn Treebank (Santorini, 1990; Marcus and Santorini (1991)), and similar efforts within the ICAME community[1], though many others have advocated the use of self-organizing re-estimation techniques that do not require access to tagged training material. The relative merits of the two approaches are discussed in Merialdo (1991); it appears that tagged text is preferable to re-estimation when a sufficient quantity of tagged text is available.[2]

2. The State of the Art

2.1 Evaluation

Most researchers report performance in the range of 95-99% ''correct'' by word, which is generally regarded by the computational linguistic community as a major advance over previous results,[3] though it is generally accepted that performance is still well below that of people.

The difference between 95% and 99% probably depends less on differences in systems and more on differences in evaluation procedures. The problem for evaluation is that we know that the computer programs aren't doing as well as people are, but if we simply measure disagreements in a double-blind experiment, we find that both the computer programs and the human judges score about 95% correct. Although it might seem that it ought to be an easy matter to decide what is right and what is wrong, it turns out that there is considerable room for disagreement. One might think that a word has one and only one possible part of speech tag, but in fact, it turns out that there are quite a number of cases where two judges will assign different tags to the same word, and even after considerable discussion, neither judge will be prepared to give in and admit that he or she had made a mistake. It may be useful to distinguish disagreements into two types: (a) legitimate differences of opinion, and (b) outright mistakes. As a general rule, most of the differences between judges tend to be differences of opinion, whereas when a computer disagrees with a judge, there is a much larger chance that the computer is simply wrong.

The problem for evaluation, then, is to find an evaluation paradigm that will distinguish legitimate differences of opinion from outright mistakes. A commonly used method, but a biased one, is to ask a human judge to look over the output from the computer and identify the outright mistakes. Of course, this

---

1. Information on the London-Lund Corpus and other corpora available though ICAME is given in the ICAME Journal, edited by Stig Johansson, Department of English, University of Oslo.
2. An interesting question, which deserves further study, is how to combine tagged and untagged text. In many practical situations, one can obtain a ''small'' sample of tagged text (e.g., 1 million words of the Brown Corpus), but one can almost always obtain a much larger sample of untagged text (e.g., 100 million words of AP newswire). We know from studies like Merialdo's that the statistics computed directly from the Brown Corpus are more useful than re-estimated versions of the same statistics. However, intuitively, it would seem that there ought to be some independent information that one should be able to make use of in the much larger sample of untagged text. The question is whether there is a practical procedure for combining estimates obtained from each sample to produce a combined set of estimates which are more useful than either in isolation.
3. See Salton *et al.* (1990) for a comparison between a statistical part of speech tagger and a more traditional part of speech tagger.

method tends to produce higher scores than with the double-blind method, because differences of opinion are no longer counted as mistakes. And admittedly, this method does introduce a bias because the human judge tends to miss some of the outright mistakes. Nevertheless, unlike the double-blind method, this evaluation procedure does tend to distinguish human-level performance from that of the computer. This is the reason that I have tended to use this evaluation method, even though it is biased and somewhat over-optimistic.

Perhaps the most important indication of success is that people are now using these statistical tagging programs in a number of different application areas including speech synthesis (Sproat, personal communication), (Liberman and Church, 1991), speech recognition (Jelinek, 1985), (Jelinek *et al.*, 1991), information retrieval (Salton *et al.*, 1990), (Croft *et al.*, 1991), sense disambiguation (Hearst, 1991), and computational lexicography (Klavans and Tzoukermann, 1990), (Church *et al.*, 1991). Apparently, these programs must be addressing some important needs of the research community or else they wouldn't be as widely cited as they are.

2.2 How Hard is the Problem?

Should we be happy with this level of performance? On the one hand, 95-99% is a lot better than we have been doing before n-gram part of speech taggers came into fashion, but on the other hand, it still means that a large fraction of sentences will contain at least one fatal error. If subsequent processing (e.g., parsing, semantic analysis) require perfect part of speech analysis, then 95% performance is clearly not nearly good enough, and probably 99% isn't either. Perhaps we need to modify these subsequent steps so they can tolerate an error rate of 1-5%. Alternatively, we may need to aim for somewhat higher levels of tagging performance than we can currently achieve.

95% might seem like it is really very good, but in fact, it may not seem like so much when we realize that some very simple methods will achieve remarkably good performance. For example, if we simply ignore the context, and just select the most likely part of speech given the word, we will achieve nearly 90% correct. Given that such a simple system will perform so well, one might ask how it is possible to report performance below 90%, although a number of systems have managed to fall well below this baseline. (A common mistake is to focus on contextual constraints and to ignore the lexical probabilities (or to quantize them too much.)) In any case, 95% may not seem so great when we realize that it means that lexical constraints get the first 90%, and that context contributes only about half of the remaining 10%.

Many people who have not worked in computational linguistics have a strong intuition that lexical ambiguity is usually not much of a problem. It is commonly believed that most words have just one part of speech, and that the few exceptions such as ‘‘table’’ are easily disambiguated by context in most cases. This intuition is largely supported by the numbers just cited. That is, most cases can be resolved without context (e.g., 90%), and that simple n-gram models of context are sufficient for more than half of the remainder.

Nevertheless, among experts in computational linguistics, it has long been felt that lexical ambiguity is a major problem; it is said that practically any content word can be used as a noun, verb or adjective, and that local context is not always adequate to disambiguate. Introductory texts are full of contrived example sentences such as:

- Time flies like an arrow.
- Flying planes can be dangerous.

where no amount of syntactic parsing will help. These example sentences are generally taken to indicate that the parser must allow for multiple possibilities, and that subsequent levels of processing (e.g., semantics, pragmatics) will be required in order to resolve the ambiguity. Unfortunately, this strategy has not worked out well in practice because it tends to ignore the lexical probabilities, which are, in fact, the

single most important set of constraints.

3. N-grams

3.1 On the Inadequacy of N-gram Models

Given the literature on ‘‘flying planes can be dangerous,’’ it may be surprising that n-gram approaches perform as well as they do. Moreover, the received wisdom in computational linguistics has held that n-gram models are necessarily inadequate for the job. This attitude probably dates back to Chomsky’s early arguments against empiricism. Recall that empiricism was at its height in the 1950s, dominating a broad set of fields ranging from psychology (behaviorism) to electrical engineering (information theory). However, interest in statistical approaches faded rather suddenly when Chomsky argued quite successfully that statistics should not play a role in his competence model.

These days, we tend to remember Chomsky’s conclusion that n-grams are not helpful, but we tend to forget the reasoning behind the arguments themselves. I suspect that Chomsky’s arguments do not apply as well to our tagging application as they do for Chomsky’s applications, and consequently, it may be worthwhile to review Chomsky’s argument and see why the conclusion might depend on the application.

> ‘‘[T]he notion ‘grammatical in English’ cannot be identified in any way with the notion ‘higher order of statistical approximation to English.’ It is fair to assume that neither sentence (1) [Colorless green ideas sleep furiously] nor (2) [Furiously sleep ideas green colorless] (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical model for grammaticalness, these sentences will be ruled out on identical grounds as equally ‘remote’ from English. Yet (1), though nonsensical, is grammatical, while (2) is not. Presented with these sentences, a speaker of English will read (1) with a normal sentence intonation, but he will read (2) ... with just the intonation pattern given to any sequence of unrelated words... Similarly, he will be able to recall (1) much more easily than (2), to learn it much more quickly, etc... Evidently, one’s ability to produce and recognize grammatical utterances is not based on notions of statistical approximations and the like. The custom of calling grammatical sentences those that ‘can occur,’ or those that are ‘possible,’ has been responsible for some confusion here... Despite the undeniable interest and importance of semantic and statistical studies of language, they appear to have no direct relevance to the problem of determining or characterizing the set of grammatical utterances. I think that we are forced to conclude that grammar is autonomous and independent of meaning, and that probabilistic models give no particular insight into some of the basic problems of syntactic structure.’’ (Chomsky 1957, pp. 15-17)

Chomsky basically argues in favor of what I might call the ‘‘competence approximation,’’ an approximation that idealizes away from preference judgments and other aspects of language that detract from the problems that he was most interested in, namely the basic problems of syntactic structure such as wh-movement and other long-distance dependencies. Obviously, this approximation simplifies matters greatly for the kinds of applications that he had in mind.

An interesting question, then, is to compare the ‘‘competence approximation’’ with other approximations that have been suggested such as the ‘‘n-gram approximation’’. Obviously, n-grams are also just an approximation; they are clearly inadequate for capturing many of the fine details that one observes in syntax, especially the kinds of long-distance dependencies that Chomsky was most interested in. Chomsky (1956) used the following example in order to make this point and demonstrate some of the limitations of n-grams:

- The man who said that S_5, is arriving today.

Note the words *man* and *is* have to agree in number, and that this agreement dependency can span over an

arbitrary number of words since S_5 could be arbitrarily long. After discussing three examples of this form, Chomsky concluded that no n-gram model can adequately model the agreement facts that he was interested in:

> ''We find that no finite-state Markov process that produces symbols with transition from state to state can serve as an English grammar. Furthermore, the particular subclass of such processes that produce n-order statistical approximations to English do not come closer, with increasing *n*, to matching the output of an English grammar.'' (Chomsky, 1956, p. 113)

Neverless, from an engineering point of view, the key question is not whether the n-gram approximation is imperfect, but rather, the key question is whether it is more or less helpful than other approximations that one might need to consider in order to get the job done. As an alternative to the n-gram approximation, one might turn to the competence approximation, and note that it too is imperfect, since it ignores some other fine details of language that we know are important, namely statistical preferences, memory limitations and other aspects of language performance. Clearly, the comparison between the various approximations depends on the application that one has in mind. If one is interested in studying the effects of long-distance dependencies in syntax (e.g., subject-verb agreement, wh-movement) as Chomsky was, then it should be fairly obvious that the competence approximation is probably more helpful than the n-gram approximation. On the other hand, if one is interested in part of speech tagging as we are here, then it should be equally obvious that the ''competence approximation'' is of little use, since it abstracts away from the lexical probabilities, which happen to be the single most useful set of constraints. Indeed, experience has shown that the competence approximation has greatly facilitated much work in theoretical syntax, and that the n-gram approximation has led to great advances in the performance of part of speech taggers.

Moreover, there have been a number of attempts to use the wrong approximation for the wrong application, which generally turn out rather disappointingly, not surprisingly. One such attempt to use the wrong approximation for the wrong application is what I might refer to, ungraciously, as the non-deterministic non-solution.

3.2 The Non-Deterministic Non-Solution

The received wisdom in computational linguistics has held that one ought to start by looking up each word in the input text in a dictionary, and then one ought to turn to a non-deterministic parser such as an augmented transition network (ATN) (Woods, 1970) to filter out those parts of speech that do not fit into one of the many resulting parse trees. It is assumed that the remaining part of speech tags should all be considered legitimate possibilities.

Unfortunately, experience has shown that this procedure tends to generate a very long list of extremely unlikely (and unhelpful) interpretations. In most applications, it really isn't very useful to flood the user with a long set of extremely implausible suggestions. If we had been evaluating our parsers with measures such as precision and recall (Salton, 1989, pp. 248-249),[4] as they do in the information retrieval literature, then we would have found that such systems produce very high recall by compromising precision. It is well-known that one can fairly easily trade-off precision for recall, but that doing so is generally of little or no value. A real solution requires both high recall as well as high precision.

The standard non-deterministic approach has been unable to obtain high precision because it has been ignoring the lexical probabilities which turn out to be the single most important source of constraint. In

4. *Precision* and *recall* are commonly used in information retrieval. Precision is the fraction of the retrieved documents that are relevant; recall is the fraction of the relevant documents that are retrieved.

contrast, statistical part of speech taggers can naturally take advantage of lexical probabilities in a way that is not easy to capture with parsers that do not make use of frequency information. Consider, for example, the word ‘‘see,’’ which is almost always a verb, but does have an archaic nominal usage as in ‘‘the Holy See.’’ For practical purposes, ‘‘see’’ should not be considered noun/verb ambiguous in the same sense as a truly polysemous word like ‘‘saw’’. Unlike the nominal usage of ‘‘see,’’ the nominal usage of ‘‘saw’’ is reasonably salient, and actually appears four times in the Tagged Brown Corpus:

and rough out with either a hack *saw* or a cutting disk in a hand power tool
to size for battens , etc. , on a table *saw* . Besides flathead bronze screws
and does not chatter the plywood like a saber *saw* . .PP When cut , the planking is clamped in place
carpenter’s tools , including a rotary power *saw* and several other pieces of power machinery that

Admittedly, the nominal usage of ‘‘saw’’ is still the minority usage; the past tense verbal usage (VBD) is found 337 times (almost 100 times more often).

It is important that the part of speech tagger have access to distinctions in saliency such as the one between ‘‘see’’ and ‘‘saw’’. In the former case, the nominal usage of ‘‘see’’ is extremely unlikely, so unlikely in fact, that not a single case has been observed in the Brown Corpus. The odds that a particular instance of ‘‘see’’ will refer to the ‘‘Holy See’’ are probably much less than 1 in 1000, given that there are almost a thousand instances of ‘‘see’’ in the Brown Corpus, and not a one refers to the ‘‘Holy See’’. On the other hand, the nominal usage of ‘‘saw’’ is simply a long-shot; the odds that a particular instance of ‘‘saw’’ will refer to a hand-saw are better than 1 in 100. It is fairly easy to model the difference between the nominal usage of ‘‘see’’ and the nominal usage of ‘‘saw’’ within the framework of a statistical part of speech tagger, which was specifically designed to take advantage of these kinds of distinctions in saliency. In contrast, non-deterministic parsers generally have no way to model these important distinctions, and consequently, they have not worked out very well in practice.

If every possibility in the dictionary must be given equal weight (as is usually the case in most non-deterministic parsing systems), then parsing is very difficult. Dictionaries tend to focus on what is possible, not on what is likely. Consider the trivial sentence, ‘‘I see a bird.’’ For all practical purposes, every word in the sentence is unambiguous. According to (Francis and Kucera, 1982), the word ‘‘I’’ appears as a pronoun (PPLS) in 5837 out of 5838 observations (˜100%), ‘‘see’’ appears as a verb in 771 out of 772 observations (˜100%), ‘‘a’’ appears as an article in 23013 out of 23019 observations (˜100%) and ‘‘bird’’ appears as a noun in 26 out of 26 observations (˜100%). However, according to Webster’s Ninth New Collegiate Dictionary (Mish *et al.*, 1983), every word is at least two-ways ambiguous. In addition to the desired assignments of tags, the first three words are listed as nouns and the last as an intransitive verb (this last possibility is generally not listed in other dictionaries). The problem, for the part of speech tagger, is to distinguish the more likely part of speech assignments from the less likely ones.

Table 1

Word	Parts of Speech	
	More Likely	Less Likely
I	pronoun	noun (letter of alphabet)
see	verb	noun (*the Holy See*)
a	article	noun (letter of alphabet)
bird	noun	verb

One might hope that these spurious assignments could be ruled out by the parser as syntactically ill-formed. Unfortunately, this has not worked out well in practice. In our trivial example, for instance, one might hope that the assignment:

- I/noun see/noun a/noun bird/noun

could be ruled out as syntactically impossible. Unfortunately, however, there is no syntactic prohibition against a sequence of nouns such as

- [NP [N city] [N school] [N committee] [N meeting]]

Thus, if the parser is going to accept the previous analysis, then there doesn't seem to be a principled reason why it shouldn't also accept the following as equally grammatical:

- [NP [N I] [N see] [N a] [N bird]]

Similarly, the parser probably also has to accept ''bird'' as an intransitive verb, since there is nothing syntactically wrong with:

- [S [NP [N I] [N see] [N a]] [VP [V bird]]]

One might now turn to semantics, pragmatics and other higher level constraints, and hope that they might exclude some of the spurious analyses. As far as I know, no such approach has ever worked very well (outside of a ''toy'' domain), and I do not hold out much hope that such an approach will work any better in the future. At some point, I think we have to accept the fact that the lexical constraints are very important, and that they cannot be modeled very effectively under the competence approximation, since the competence approximation abstracts away from just the kinds of preferences that we need to model. The important point is that the spurious part of speech assignments aren't wrong; they are just extremely improbable.

In summary, this example illustrates three problems with the non-deterministic non-solution. First, it is not helpful to return a long list of low precision suggestions including such gems as:

- I/noun see/noun a/noun bird/noun
- I/noun see/noun a/noun bird/verb

Secondly, it is important to make use of lexical constraints when they are available. It is especially important to make use of them when they are overwhelming, as they are in this example. The word ''see'' is almost unambiguous, unlike a word like ''saw,'' where both the noun (a hand-saw) and the verb (past tense of ''see'') are reasonably salient.

Thirdly, one should not expect contextual constraints to contribute very much. It has been my experience that contextual constraints are much weaker than lexical constraints. Parsing is a very difficult business (as we have all learned the hard way), and most parsers with reasonable coverage will accept the good with the bad. It is unlikely that such a broad-coverage parser could be expected to recover from inadequacies in a lexical model that ignores the difference between almost unambiguous words like ''see'' and truly polysemous words like ''saw.''

Before discussing the details of the tagging algorithm, I would like to say a few words about how I came to be convinced that it was time to question the received wisdom. A few years ago, I needed a few slides for a tutorial on chart parsing and so I quickly wrote a chart parser with a ''toy'' grammar and a ''toy'' lexicon, very much like what can be found on pp. 269-270 of (Martin *et al.*, 1987). Just before I finalized the slides, though, I decided to replace the ''toy'' lexicon with Webster's Seventh New Collegiate Dictionary, which happened to be readily available in computer readable form. The first sentence I tried was, ''I saw a bird,'' and to my surprise it came out with more than one interpretation. Apparently, this simple example was much too complicated for my tutorial. After realizing that ''saw'' was a poor choice because of the

polysemy, I then decided to pick a simple ‘‘unambiguous’’ word like ‘‘see’’. But unfortunately, I quickly found out that ‘‘I see a bird’’ is no simpler than ‘‘I saw a bird’’. At this point, I became convinced that (1) I needed a model of the lexicon that would distinguish an almost unambiguous word like ‘‘see’’ from a truly polysemous word like ‘‘saw,’’ and that (2) it was hopeless to expect the parser to magically extract the missing information from context, when the key distinction is a lexical matter, and has very little to do with context.

4. The Currently Accepted Practice

4.1 An Example

Consider once again the trivial sentence, ‘‘. . I see a bird . .’’ (We will not discuss the front end prepass which tokenizes the input sentence into words and pads sentences with two special tokens at each end.) The problem is to find an assignment of parts of speech to words that optimizes both lexical probabilities $Prob(p_i|w_i)$ and contextual probabilities $Prob(p_i|p_{i+1}p_{i+2})$, both of which are estimated from the Tagged Brown Corpus. Conceptually, enumerate all sequences of parts of speech $\{p_i\}$ that could correspond to the input sentence of N words $\{w_i\}$. Then score each sequence by:

$$\underset{\{p_i\}}{MAX} \prod_{i=1}^{N} \frac{Prob(p_i|w_i)\,Prob(p_i|p_{i+1}p_{i+2})}{Prob(p_i)}$$

and select the highest scoring sequence. (We will not discuss the normalization factor, $Prob(p_i)$, which compensates for the fact that $Prob(p_i)$ is counted twice in the numerator, once in the lexical probabilities and once in the contextual probabilities.)

The ‘‘I see a bird’’ example is illustrated in Table 2.

====== INSERT Table 2 ABOUT HERE ======

There are 8 possible part of speech sequences, labeled A1 through A8, duplicated just below for convenience.

Table 3

	.	.	I	see	a	bird	.	.
A1	.	.	PPSS	VB	AT	NN	.	.
A2	.	.	PPSS	VB	IN	NN	.	.
A3	.	.	PPSS	UH	AT	NN	.	.
A4	.	.	PPSS	UH	IN	NN	.	.
A5	.	.	NP	VB	AT	NN	.	.
A6	.	.	NP	VB	IN	NN	.	.
A7	.	.	NP	UH	AT	NN	.	.
A8	.	.	NP	UH	IN	NN	.	.

That is, the pad token ‘‘.’’ has just one part of speech: ‘‘.’’; the word ‘‘I’’ can be either a pronoun PPSS or a proper noun NP (according to (Francis and Kucera, 1982)); the word ‘‘see’’ is either an uninflected verb VB or an interjection UH; ‘‘a’’ is either an article AT or a preposition IN (from French); ‘‘bird’’ is always a noun NN. Note that the part of speech choices given in (Francis and Kucera, 1982) are somewhat different from the ones discussed previously, which were given in Mish *et al.*, 1983). Most of the remainder of this discussion will work with the choices in (Francis and Kucera, 1982).

In Table 2, each part of speech sequence is followed by two two rows of numbers corresponding to the lexical and contextual probabilities. The probabilities for A1 are repeated here:

Table 4

A1	.	.	PPSS	VB	AT	NN	.	.
lex	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
con	0.99	0.20	0.07	0.07	0.23	0.25	1.00	1.00

The lexical probabilities are all nearly 1. In other words, it is very likely that ‘‘I’’ is a pronoun, ‘‘see’’ is a verb, ‘‘a’’ is an article and ‘‘bird’’ is a noun. The contextual probabilities are also quite large. The third number, 0.07, for example, denotes the probability of finding a pronoun before a verb and and article. It was computed by dividing the number of PPSS, VB, AT trigrams in the training corpus by the number of VB, AT bigrams.

The bottom line score for A1 is 10^{-4}, which is computed by multiplying all of the lexical and contextual probabilities. As one can see from the right most column in Table 2, A1 has a much higher score than any other part of speech sequence; A1 (pronoun, verb, article, noun) scores 10,000 times better than A2 (pronoun, verb, preposition, noun), the next best sequence.

4.2 The Dynamic Programming Solution

In this example, there are eight possible sequences since there were three 2-ways ambiguous tokens (‘‘I’’, ‘‘see,’’ and ‘‘a’’), and $2^3 = 8$ paths. In general, let us assume that words are no more than k ways ambiguous, where k is about 10, and that input sentences are no more than N words long, where N is about 100. Then, if the search were to literally enumerate all part of speech sequences, it might need to look at $k^N (\approx 10^{100})$ part of speech sequences. Fortunately, there is a dynamic programming solution to the search since the scoring function can see only 2 words away.

Suppose (for convenience only), we start the search from the end of the sentence. First we consider the possible part of speech sequences for ‘‘bird’’:

0.25 bird/NN ./. ./.

The score, 0.25, was computed by multiplying the lexical probability for NN given ‘‘bird’’ (1.00) and the contextual probability of NN given the following two parts of speech (0.25). We now consider the possible part of speech sequences for ‘‘a’’:

0.06 a/AT bird/NN ./. ./.
3×10^{-6} a/IN bird/NN ./. ./.

The score, 0.06, was computed by multiplying the previous score (0.25) with the lexical probabilities for AT given ‘‘a’’ (1.00) and the contextual probability for AT given the next two parts of speech (0.23). We now consider the possible part of speech sequences for ‘‘see’’: (The score for the last two paths is actually slightly more than 0, but we won’t deal with that here.)

4×10^{-3} see/VB a/AT bird/NN ./. ./.
1×10^{-7} see/VB a/IN bird/NN ./. ./.
0 see/UH a/AT bird/NN ./. ./.
0 see/UH a/IN bird/NN ./. ./.

Now, find assignments of ‘‘I’’ and score. Note, however, that it is no longer necessary to hypothesize that ‘‘a’’ might be a French preposition IN because all four paths, ‘‘I/PPSS see/VB a/IN bird/NN,’’ ‘‘I/NP see/VB a/IN bird/NN,’’ ‘‘I/PPSS see/UH a/IN bird/NN’’ and ‘‘I/NP see/UH a/AT bird/NN’’ score no better than some other path and there is no way that any additional input could make any difference. In particular,

the path, ‘‘I/PPSS see/VB a/IN bird/NN’’ scores no better than the path ‘‘I/PPSS see/VB a/AT bird/NN,’’ and additional input will not help ‘‘I/PPSS see/VB a/IN bird/NN’’ because the contextual scoring function has a limited window of three parts of speech, which is not enough to see past the existing ‘‘I/PPSS’’ and ‘‘see/VB.’’

10^{-4}	I/PPSS	see/VB	a/AT	bird/NN ./.	./.
10^{-9}	I/NP	see/VB	a/AT	bird/NN ./.	./.
0	I/PPSS	see/UH	a/AT	bird/NN ./.	./.
0	I/NP	see/UH	a/AT	bird/NN ./.	./.

The search continues two more iterations for the two pad characters and ultimately concludes that part of speech sequence A1 is the best. Because of the dynamic programming optimization, only $k^2 (\approx 10^2)$ paths need to be kept around as each of the N input words are processed. Thus, the dynamic programming optimization reduces the search space from k^N down to only Nk^2.[5]

5. Smoothing Issues

5.1 Smoothing Lexical Probabilities

Some of the probabilities are very hard to estimate by direct counting because of Zipf’s Law (frequency is roughly proportional to inverse rank). Consider, for example, the lexical probabilities. We need to estimate how often each word appears with each part of speech. Unfortunately, because of Zipf’s Law, no matter how much text we look at, there will always be a large tail of words that appear only a few times. In the Brown Corpus, for example, 40,000 words appear five times or less. If a word such as ‘‘yawn’’ appears once as a noun and once as a verb, what is the probability that it can be an adjective? It is impossible to say without more information.

In general, there is a very serious sparse data problem. If there are 40 different parts of speech, then we need to estimate 40 parameters for each of these 40,000 words. At best, we will have 5 instances for each word, so will have 5 data points in order to estimate 40 parameters. The true situation is actually worse than this because many of the 40,000 words appear less than 5 times, so we actually have less than 5 data points in many cases. One might think that we could fix the sparse data problem by merely collecting more data, but ironically, if we do that we find that the sparse data problem actually gets worse. The problem is that the number of words that appear only a few times increases as we collect more and more data. As we collect more and more data, we begin to learn more and more just how little we actually know.[6]

In some cases, conventional dictionaries can be used to help alleviate the sparse data problem. For example, the fact that none of our dictionaries indicate that *yawn* is an adjective should make us feel more comfortable in assigning it a small probability, perhaps even zero probability. Church (1988) describes an

---

5. This dynamic programming algorithm finds the best path through the lattice subject to the optimization criterion. Of course, it is possible to use an N-best search algorithm (e.g., Soong and Huang (1990), Schwartz and Austin (1990)) in order to find the best N solutions. de Marcken (1990) took a similar approach and reported several choices under certain conditions. My problem with these non-deterministic approaches to part of speech tagging is that I believe very strongly that the lexical constraints are superior to whatever constraints one is likely to find at higher levels of processing, and thus, I believe it doesn’t do any good to postpone the decision. It seems to me that those who advocate postponing the decision need to show that there is some advantage in doing so. Thus far, the argument has tended to rest mainly on showing improved recall, but the argument tends to omit the fact that the improvement is achieved by merely trading off precision for recall, which is of little or no value for reasons discussed previously.

6. It is commonly believed that the vocabulary of English (or any other natural language) is relatively small and fixed (e.g., 50,000 words). In fact, over all samples of text that I have been able to look at (currently numbering into the hundreds of millions of words), it appears that the vocabulary continues to grow at a very steady rate. The growth rate is not quite linear, but it appears to be more than $\sqrt{N}$. The constants depend on what you count as a word and the source of the text, among other things. The basic shape of the growth rate curve, though, seems to be fairly clear, and does not depend too much on these details.

ad hoc procedure for combining evidence from dictionaries with evidence obtained from tagged corpora. I would like to report that I have replaced this *ad hoc* method (add one to the relevant counts) with something more principled such as deleted interpolation (Jelinek and Mercer, 1985) or the Good-Turing method (Good, 1953). Unfortunately, I have to report that I still use the same *ad hoc* methods reported in Church (1988), even though I have learned that those *ad hoc* methods are, in fact, much worse than I had thought (Church and Gale, 1990). It is very possible that my feeble attempt to grapple with the sparse data problem may have actually made matters worse rather than better.

It turns out that it is often non-trivial to combine evidence collected from different sources (such as dictionaries and tagged corpora) because almost every source uses their own unique tagging system. One can always hand-tune a set of mapping rules that convert one tagging system into another, but one invariably suspects that the mapping rules might be introducing more trouble than they are worth. At the very least, the mapping rules require that one make use of morphological regularities since most dictionaries do not list morphological variants.

5.2 Smoothing Across Morphological Variants

One would hope that it would be possible to predict the lexical probabilities by making use of what we know about morphologically related forms. Even this apparently benign step is fraught with peril. One might think that it would be fairly safe to assume that adding an 's' to the end of a word would not change its lexical probabilities very much. Unfortunately, even this seemingly innocuous move can lead to serious difficulties. Table 5 lists a number of words in two forms: (1) the base form, and (2) the base form with an extra 's' added to the end. In each case, I counted the number of times that the form was found in the Tagged Brown Corpus as a noun (e.g., nn, nns), and the number of times that it was found with some other part of speech (e.g., vb, vbz, jj). I am particularly interested in the fourth and seventh columns. One might hope that morphologically related forms would share the same probability of being a noun. That is, one might hope that the fourth and seventh columns would be highly correlated. Apparently, there is a correlation, but it is much weaker than I had hoped (0.42). A scatter plot of the fourth and seventh columns are presented in Figure 1.[7] Note that there is a trend for morphological variants to all share more or less the same probability of being a noun, but there are plenty of very striking exceptions.

====== INSERT Table 5 ABOUT HERE ======

====== INSERT Figure 1 ABOUT HERE ======

Perhaps the approach outlined in the previous paragraph is just too simple-minded. It may need to be enhanced in a number of ways, both linguistic and statistical. In particular, some of the worst outliers in Figure 1 are adjective forms (e.g., *moderate, parallel, close*). It may be necessary to treat adjectives separately since adjective morphology is different from noun-verb morphology. In addition, the argument is also taking ratios of frequency counts. It is well-known that ratios of small counts are extremely unstable. Again, it may be necessary to take some more precautions. Nevertheless, despite these problems, I believe the argument does a fairly good job in illustrating just how hard it can be to predict lexical probabilities on the basis of morphological variants.

---

7. Points are used to indicate the location of words in the upper right hand corner of Figure 1 because there are so many words in that corner.

5.3 Smoothing Contextual Probabilities

The lexical probabilities are not the only probabilities that require smoothing. The contextual probabilities also raise some interesting estimation questions. They too tend to have a very skewed distribution, and consequently, even after looking at a very large training corpus, there will still be many n-grams that have not been observed. It is clear that the contextual frequencies require smoothing. Zeros should be avoided.

Nevertheless, I have relatively little to say about estimating the contextual probabilities for three reasons: (1) it relatively well-studied,[8] (2) it is relatively easy since there are relatively few parameters to estimate,[9] and (3) it is relatively unimportant since the contextual probabilities don't matter very much compared with the lexical probabilities.

The third point, of course, dominates the argument. I suspect that most of the gains that we are likely to see in the near future will be in finding better ways to estimate lexical probabilities. We tend to focus our attention on contextual constraints because we have all been trained to think about grammatical questions, but in fact, I am fairly convinced that lexical constraints are much more important, and much more difficult to model for a number of reasons including some of the ones mentioned in the previous subsection. However, I am confident that we will be able to make progress in this area, given the massive amount of recent interest in both part of speech tagging in specific, and data collection in general.

6. Conclusion

As a result of corpus collection efforts such as (Francis and Kucera, 1982) and related data-collection efforts around the world, quite a number of part of speech taggers have been developed including (e.g., Leech *et al.* (1983), Jelinek (1985), Deroualt and Merialdo (1986), Church (1988), DeRose (1988), Kupiec (1989), Ayuso *et al.* (1990), de Marcken (1990), Boggess *et al.* (1991), Merialdo (1991)). These programs, which are based on probabilities derived from corpus data, are extremely successful; they work on unrestricted texts, with reasonable accuracy and efficiency. Performance is very encouraging as can be seen from the Appendix. More importantly, these programs are now beginning to be used in quite a number of applications areas including speech synthesis (Sproat, personal communication), (Liberman and Church, 1991), speech recognition (Jelinek, 1985), (Jelinek *et al.*, 1991), information retrieval (Salton *et al.*, 1990), (Croft *et al.*, 1991), sense disambiguation (Hearst, 1991), and computational lexicography (Klavans and Tzoukermann, 1990), (Church *et al.*, 1991).

Much of this progress has been possible because lexical probabilities (probability of a part of speech given the word) are now being estimated directly from corpus data, and can therefore be estimated much more accurately than before. Until recently, it had been common practice for most researchers in computational linguistics to concentrate their energies on contextual constraints, which are probably much less important than lexical constraints.

There is, of course, much room for improvement. I suspect models of lexical probabilities will continue to

---

8. See Katz (1987) for a discussion of how to ''back-off'' from trigram estimates to bigram and unigram estimates, when appropriate. See Ayuso *et al.* (1990) and Merialdo (1991) for some discussion on how much training text is required in order to obtain acceptable results.

9. If there are 40 parts of speech, then there are 40^3 or 64,000 parameters to estimate. If there are a million words of training material, then there are more than 15 data points for each parameter (on average). Obviously, the data points are generally not uniformly distributed, but nevertheless, the situation with the contextual probabilities is considerably better than with the lexical probabilities. In that case, there are about 50,000 words in the Brown Corpus with 40 parts of speech each, so there are 2.5 million parameters to estimate. Given that we have only 1 million words of training data, we already have more parameters than data points. And to make matters worse, the data are not well-distributed.

improve, especially as we continue to collect more and more corpus evidence and we find ways to make better and better use of the evidence that we have. It is particularly important that we find ways to combine evidence obtained from different sources using different notational systems. A major problem, here, is to find a reliable way of making inferences across morphological variants. One would hope that it would be possible to predict the lexical probabilities by making use of what we know about morphologically related forms, but as we have seen, even this apparently innocuous move can lead to serious difficulties. Nevertheless, I am convinced that we will continue to see progress on these questions given the level of effort that the research community is currently devoting to these problems.[10]

10. There will also been improvements in the modeling of contextual constraints, given the current level of effort that is being directed in that direction. I am less confident, however, that this work will produce significant improvements in performance, since I believe that performance is currently limited by lexical issues (as opposed to contextual ones), and that this situation is unlikely to change for some time to come.

References

[1] Ayuso, D., Bobrow, R., MacLaughlin, D., Meteer, M., Ramshaw, L., Schwartz, R., Weischedel, R. (1990) ‘‘Toward Understanding Text with a Very Large Vocabulary,’’ DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 354-358.

[2] Boggess, L., Agarwal, R., Davis, R. (1991), ‘‘Disambiguation of Prepositional Phrases in Automatically Labelled Technical Text,’’ AAAI, pp. 155-159.

[3] Chomsky, N., ‘‘Three Models for the Description of Language,’’ IRE Transactions on Information Theory, vol. IT-2, Proceedings of the Symposium on Information Theory, 1956.

[4] Chomsky, N. (1957) *Syntactic Structures*, The Hague: Mouton & Co.

[5] Church, K. (1988) ‘‘A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text,’’ Second Conference on Applied Natural Language Processing, Austin, Texas.

[6] Church, K., and Gale, W. (1990) ‘‘Poor Estimates of Context are Worse than None,’’ DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 283-287.

[7] Church, K., Gale, W., Hanks, P., Hindle, D. (1991) ‘‘Using Statistics in Lexical Analysis,’’ in Zernik, U. (ed.) ‘‘Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon,’’ Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[8] Croft, W., Turtle, H., Lewis, D. (1991) ‘‘The Use of Phrases and Structured Queries in Information Retrieval,’’ in Bookstein, A., Chiaramella, Y., Salton, G., Raghavan, V. (eds.), SIRIR Forum.

[9] de Marcken, C. (1990) ‘‘Parsing the LOB Corpus,’’ Association for Computational Linguistics, pp. 243-251.

[10] DeRose, S., ‘‘Grammatical Category Disambiguation by Statistical Optimization,’’ Computational Linguistics, Vol. 14, No. 1, 1988.

[11] Deroualt, A. and Merialdo, B. (1986) ‘‘Natural Language Modeling for Phoneme-to-text Transcription,’’ *IEEE Transations on Pattern Analysis and Machine Intelligence* PAMI-8(6), pp. 742-749.

[12] Francis, W., and Kucera, H., ‘‘Frequency Analysis of English Usage,’’ Houghton Mifflin Company, Boston, 1982.

[13] Good, I. (1953) ‘‘The Population Frequencies of Species and the Estimation of Population Parameters,’’ *Biometrika*, vol. 40, pp. 237-264.

[14] Hearst, M. (1991) ‘‘Toward Noun Homonym Disambiguation Using Local Context in Large Text Corpora,’’ in *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, available from the UW Centre for the New OED and Text Research, University of Waterloo, Waterloo, Ontario, Canada.

[15] Heidorn, G., Jensen, K., Miller, L., Byrd, F. and Chodorow, M. (1982) ‘‘The EPISTLE Text Critiquing System,’’ *IBM Systems Journal*, 21:3, pp. 305-326.

[16] Hindle, D. (1989) ‘‘Acquiring Disambiguation Rules from Text,’’ ACL, pp. 118-125.

[17] Jelinek, F. (1985) ‘‘Self-organized Language Modeling for Speech Recognition,’’ IBM Report, also available in Waibel, A. and Kai-Fu, L. (eds.) (1990) *Readings in Speech Recognition*, Morgan Kaufmann Publishers, San Mateo, California, pp. 450-506.

[18] Jelinek, F. and Mercer, R. (1985) ‘‘Probability Distribution Estimation From Sparse Data,’’ *IBM Technical Disclosure Bulletin*, Vol. 28, pp. 2591-2594.

[19] Jelinek, F., Mercer, R. and Roukos, S. (1991) ‘‘Principles of Lexical Language Modeling for Speech Recognition,’’ in Furui, S. and Mohan, M. (eds.), ‘‘Advances in Speech Signal

Processing,'' Marcel Dekker, New York.

[20] Karlsson, F. (1990) ''Constraint Grammar as a Framework for Parsing Running Text,'' COLING, pp. 168-173.

[21] 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*, v. ASSP-35, pp. 400-401.

[22] Klavans, J. and Tzoukermann, E. (1990) ''Linking Bilingual Corpora and Machine Readable Dicitonaries with the BICORD System,'' in *Proceedings of the Sixth Annual Conference of the UW Centre for the New OED and Text Research*, available from the UW Centre for the New OED and Text Research, University of Waterloo, Waterloo, Ontario, Canada.

[23] Kupiec, J. (1989) ''Augmenting a Hidden Markov Model for Phrase-Dependent Word Tagging,'' DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 92-98.

[24] Leech, G., Garside, R., Atwell, E., ''The Automatic Grammatical Tagging of the LOB Corpus,'' ICAME News 7, 13-33, 1983.

[25] Liberman, M., and Church, K. (1991) ''Text Analysis and Word Pronunciation in Text-to-Speech Synthesis,'' in Furui, S. and Mohan, M. (eds.), ''Advances in Speech Signal Processing,'' Marcel Dekker, New York.

[26] Marcus, M. and Santorini, B. (1991) ''Building Very Large Natural Language Corpora: the Penn Treebank,'' unpublished ms., University of Pennsylvania.

[27] Martin, W., Church, K., and Patil, R. (1987) ''Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and Experimental Results,'' Bolc, L. (ed.), ''Natural Language Parsing Systems,'' Springer-Verlag.

[28] Merialdo, B. (1991) ''Tagging Text with a Probabilistic Model,'' IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[29] Meteer, M., Schwartz, R., Weischedel, R. (1991), ''Studies in Part of Speech Labelling,'' DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 331-336.

[30] Mish *et al.* (1983) ''Webster's Ninth New Collegiate Dictionary,'' Merriam Company, Springfield, Massachusetts.

[31] Salton, G. (1989) *Automatic Text Processing*, Addison-Wesley Publishing Co.

[32] Salton, G., Zhao, Z., Buckley, C. (1990) ''A Simple Systactic Approach for the Generation of Indexing Phrases,'' TR90-1137, Department of Computer Science, Cornell University.

[33] Santorini, B. (1990) ''Part-of-speech tagging guidelines for the Penn Treebank Project,'' Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

[34] Schwartz, R. and Austin, S. (1990) ''Efficient, High-Performance Algorithms for N-Best Search,'' DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 6-11.

[35] Sinclair, J., Hanks, P., Fox, G., Moon, R., Stock, P. et al. (eds.) (1987) *Collins Cobuild English Language Dictionary*, Collins, London and Glasgow.

[36] Soong, F. and Huang, E. (1990) ''A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition,'' DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, San Mateo, California, pp. 12-19.

[37] Woods, W. (1970), ''Transition Network Grammars for Natural Language Analysis,'' CACM 13:10.

Appendix: Sample Results

The following news item was distributed over the Associated Press newswire on October 24, 1991. The part of speech tags and the noun phrase brackets were inserted by the program described in (Church, 1988). Note that the news item contains several typos, although they do not seem to cause the part of speech tagging program too much trouble. The tagging program introduces at least four part of speech errors. The algorithm for inserting brackets is not discussed here because it is somewhat peripheral to the main topic of this paper, and also because it does not work as well as the tagging algorithm.

[International/NP/NP Business/NP/NP Machines/NP/NP Corp/NP./NP] sued/VBD [a/AT leading/VBG marketer/NN] of/IN [computers/NNS Thursday/NP] ,/, claiming/VBG [it/PPO] cannibalize/VB [IBM/NP/NP mainframes/NNS] ,/, [counterfeits/NN parts/NNS] and/CC sells/VBZ or/CC leases/VBZ [them/PPO] to/TO [unsuspecting/JJ customers/NNS]

[The/AT lawsuit/NN] against/IN [Comdisco/NP/NP Inc/NP./NP] marked/VBD [IBM/NP/NP ’s/$ second/CD major/JJ legal/JJ action/NN] against/IN [another/DT computer/NN company/NN] in/ININ [the/AT past/JJ few/JJ weeks/NNS] ./. [It/PPS] also/RB stood/VBD in/ININ [stark/JJ contrast/NN] to/TOIN [the/AT friendly/JJ strategic/JJ alliances/NNS IBM/NP/NP] has/HVZ struck/VBN recently/RB with/IN [some/DTI competitors/NNS] such/JJ as/CS [Apple/NP/NP Computer/NP/NP Inc/NP./NP] and/CC [Wang/NP/NP Laboratories/NP/NP Inc/NP/NP] ./.

‘‘/‘‘ [The/AT gloves/NNS] are/BER off/IN ,/, ’’/’’ said/SAIDVBD [Sam/NP/NP Albert/NP/NP] ,/, [an/AT independent/JJ computer/NN consultant/NN] and/CC [former/JJ IBM/NP/NP software/NN executive/NN] ./. ‘‘/‘‘ [I/PPSS] think/VB [the/AT IBM/NP/NP company/NN] ,/, unlike/IN in/ININ [past/JJ decades/NNS] when/WRB [it/PPS] was/BEDZ viewed/VBN as/CS [a/AT monopolist/NN] ,/, is/BEZ now/RB viewed/VBN as/CS [an/AT underdog/NN] ./. [They/PPSS] ’re/BER losing/VBG [market/NN share/NN] ./. [They/PPSS] ’re/BER fighting/VBG for/FORIN [their/JJ right/NN] ./. ’’/’’

[The/AT world/NN ’s/$ biggest/JJ computermaker/NN] has/HVZ been/BEN suffering/VBG through/IN [an/AT earnings/NNS] slump/VB while/CS trying/VBG to/TO protect/VB [the/AT trade/NN secrets/NNS] [that/WPS] historically/RB have/HV given/VBN [it/PPO] [a/AT dominant/JJ edge/NN] ./.

[The/AT IBM/NP/NP suit/NN] against/IN [Comdisco/NP/NP] ,/, filed/VBN in/ININ [federal/JJ court/NN] in/ININ [Chicago/NP] ,/, claimed/VBD [the/AT Rosemont/NP/NP] ,/, [Ill/NP./NP] ,/, based/VBN [company/NN] was/BEDZ secretly/RB disassembling/VBG [IBM/NP/NP mainframes/NNS] ,/, counterfeiting/VBG [memory/NN cards/NNS] and/CC reselling/VBG or/CC leasing/VBG [them/PPO] ./. [The/AT suit/NN] alleges/VBZ [the/AT parts/NNS] don’t/DO comply/VB with/IN [IBM/NP/NP standards/NNS] and/CC are/BER likely/JJ to/TO fail/VB ./.

‘‘/‘‘ [This/DT] is/BEZ [an/AT outrage/NN] and/CC must/MD be/BE stopped/VBN ,/, ’’/’’ [William/NP/NP O/NP./NP Grabe/NP/NP] ,/, [IBM/NP/NP vice/NN president/NN] and/CC [general/JJ manager/NN] of/IN [the/AT company/NN ’s/$ domestic/JJ marketing/NN] and/CC [services/NNS division/NN] ,/, said/SAIDVBD in/ININ [a/AT press/NN statement/NN] on/ONIN [the/AT suit/NN] ./.

[John/NP/NP Reilly/NP/NP] ,/, [a/AT spokesman/NN] for/FORIN [IBM/NP/NP] ,/, said/SAIDVBD [the/AT company/NN] was/BEDZ seeking/VBG [unspecified/JJ monetary/JJ damages/NNS] and/CC [a/AT court/NN order/NN] to/TO halt/VB [the/AT alleged/VBN actions/NNS]

[Telephone/NN requests/NNS] for/FORIN [Comdisco/NP/NP comment/NN] on/ONIN [the/AT suit/NN] were/BED referred/VBN to/TOIN [Philip/NP/NP Hewes/NP/NP] ,/, [the/AT company/NN ’s/$ general/JJ counsel/NN] ,/, [who/WPS] did/VBD not/RB [return/NN two/CD calls/NNS] by/IN [late/JJ Thursday/NP afternoon/NN] ./. [He/PPS] was/BEDZ said/SAIDVBD by/IN [a/AT secretary/NN]

to/TO be/BE in/ININ [a/AT meeting/NN] ./.

[The/AT IBM/NP/NP suit/NN] claims/VBZ [the/AT company/NN] ‘‘/‘‘ has/HVZ lost/VBN [the/AT ability/NN] to/TO control/VB [the/AT quality/NN] of/IN [the/AT memory/NN products/NNS] marketed/VBN under/IN [its/JJ trademark/NN] ,/, and/CC [its/JJ goodwill/NN] and/CC [reputation/NN] have/HV been/BEN imperiled/VBN ./. ’’/’’

[The/AT company/NN] also/RB said/SAIDVBD [it/PPS] was/BEDZ ‘‘/‘‘ continuing/VBG to/TO investigate/VB [facts/NNS] surrounding/VBG [other/JJ third/CD parties/NNS] [who/WPS] may/MD be/BE passing/VBG off/IN [counterfeit/JJ memory/NN cards/NNS] as/CS [genuine/JJ IBM/NP/NP parts/NNS] ’’/’’

[IBM/NP/NP] ,/, headquartered/VBN in/ININ [Armonk/NP/NP] ,/, [N.Y/NP./NP] ,/, is/BEZ [the/AT biggest/JJ producer/NN] of/IN [mainframe/NN computers/NNS] ,/, [the/AT roomsized/VBN machines/NNS] that/CS [large/JJ businesses/NNS] use/VB to/TO store/VB and/CC process/VB [vast/JJ quantites/NNS] of/IN [information/NN] ./.

[The/AT company/NN] recently/RB rolled/VBD out/IN [a/AT new/JJ generation/NN] of/IN [mainframes/NNS] ,/, [which/WDT] account/VB for/FORIN about/IN [half/ABN its/JJ earnings/NNS] when/WRB associated/VBN [software/NN] and/CC [equipment/NN] is/BEZ included/VBN ./. [Last/JJ week/NN] ,/, [it/PPS] reported/VBD [an/AT 85/CD percent/NN drop/NN] in/ININ [third-quarter/JJ earnings/NNS] ,/, largely/RB due/JJ to/TOIN [the/AT effects/NNS] of/IN [the/AT recession/NN] ./.

On/ONIN [Oct/NP./NP 11/CD] ,/, [IBM/NP/NP] sued/VBD [Seagate/NP/NP Technology/NP/NP Inc/NP./NP] ,/, [a/AT leading/VBG maker/NN] of/IN [magnetic/JJ disk/NN] drives/VBZ for/FORIN [computers/NNS] ,/, claiming/VBG [the/AT company/NN] and/CC [a/AT former/JJ IBM/NP/NP product/NN development/NN manager/NN] had/HVD stolen/VBN [trade/NN secrets/NNS] ./. [The/AT suit/NN] ,/, filed/VBN in/ININ [federal/JJ court/NN] in/ININ [Minneapolis/NNS] ,/, also/RB accused/VBN [Seagate/NP/NP] of/IN trying/VBG to/TO recruit/VB [other/JJ IBM/NP/NP employees/NNS] to/TO acquire/VB [trade/NN secrets/NNS] ./.

[Officials/NNS] at/IN [Seagate/NP/NP] ,/, headquartered/VBN in/ININ [Scotts/NP/NP Valley/NP/NP] ,/, [Calif/NP./NP] ,/, have/HV not/RB responded/VBN to/TOIN [telephone/NN requests/NNS] for/FORIN [comment/NN] on/ONIN [the/AT suit/NN] ./.

Table 2

	.	.	I	see	a	bird	.	.	
A1	.	.	PPSS	VB	AT	NN	.	.	
lex	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
con	0.99	0.20	0.07	0.07	0.23	0.25	1.00	1.00	10^{-4}
A2	.	.	PPSS	VB	IN	NN	.	.	
lex	1.00	1.00	1.00	1.00	10^{-4}	1.00	1.00	1.00	
con	0.99	0.20	0.08	0.03	0.13	0.25	1.00	1.00	10^{-9}
A3	.	.	PPSS	UH	AT	NN	.	.	
lex	1.00	1.00	1.00	10^{-3}	1.00	1.00	1.00	1.00	
con	0.99	1.00	0.00	0.00	0.23	0.25	1.00	1.00	0
A4	.	.	PPSS	UH	IN	NN	.	.	
lex	1.00	1.00	1.00	10^{-3}	10^{-4}	1.00	1.00	1.00	
con	0.99	1.00	0.00	0.00	0.13	0.25	1.00	1.00	0
A5	.	.	NP	VB	AT	NN	.	.	
lex	1.00	1.00	10^{-4}	1.00	1.00	1.00	1.00	1.00	
con	0.97	0.03	0.01	0.07	0.23	0.25	1.00	1.00	10^{-10}
A6	.	.	NP	VB	IN	NN	.	.	
lex	1.00	1.00	10^{-4}	1.00	10^{-4}	1.00	1.00	1.00	
con	0.97	0.03	0.01	0.03	0.13	0.25	1.00	1.00	10^{-15}
A7	.	.	NP	UH	AT	NN	.	.	
lex	1.00	1.00	10^{-4}	10^{-3}	1.00	1.00	1.00	1.00	
con	0.97	0.00	0.00	0.00	0.23	0.25	1.00	1.00	0
A8	.	.	NP	UH	IN	NN	.	.	
lex	1.00	1.00	10^{-4}	10^{-3}	10^{-4}	1.00	1.00	1.00	
con	0.97	0.00	0.00	0.00	0.13	0.25	1.00	1.00	0

Table 5

Word	**Base Form**			**Base Form + s**		
	noun	non–noun	%	noun	non–noun	%
abuse	13	3	81	7	0	100
account	90	27	77	27	9	75
ban	6	1	86	0	1	0
bar	68	3	96	36	3	92
brief	9	61	13	1	0	100
care	85	75	53	1	7	12
center	175	12	94	45	7	87
close	15	215	7	0	6	0
cost	179	44	80	160	12	93
cup	43	1	98	14	0	100
display	28	12	70	15	6	71
drain	11	4	73	4	1	80
dump	1	3	25	1	0	100
end	360	40	90	54	13	81
fleet	11	1	92	1	0	100
force	176	24	88	142	6	96
front	142	75	65	7	0	100
host	36	0	100	5	0	100
interview	31	1	97	17	0	100
issue	130	14	90	63	3	95
lot	123	1	99	38	1	97
moderate	0	21	0	3	1	75
motor	52	1	98	7	0	100
note	73	33	69	50	4	93
paragraph	10	0	100	14	0	100
parallel	10	30	25	1	1	50
phone	47	5	90	5	2	71
pledge	2	0	100	3	0	100
premise	7	0	100	8	0	100
quote	2	15	12	4	0	100
rank	20	2	91	19	1	95
report	115	38	75	65	13	83
resort	6	6	50	1	0	100
revolt	8	0	100	2	0	100
root	25	3	89	21	0	100
section	132	0	100	65	0	100
share	56	39	59	41	4	91
speed	75	5	94	13	1	93
spread	17	62	22	1	9	10
trip	80	1	99	29	0	100
view	165	18	90	49	2	96
vote	49	26	65	17	3	85
warrant	9	11	45	5	0	100
wonder	26	38	41	6	2	75