Where the Tagger Falters

Elliott Macklovitch
Canadian Workplace Automation Research Centre
1575 Chomedey Boulevard
Laval, Quebec
Canada H7V 2X2

Abstract

Statistical n-gram taggers like that of [Church 1988] or [Foster 1991] assign a part-of-speech label to each word in a text on the basis of probability estimates that are automatically derived from a large, already tagged training corpus. This paper examines the grammatical constructions which cause such taggers to falter most frequently. As one would expect, certain of these errors are due to linguistic dependencies that extend beyond the limited scope of statistical taggers, while others can be seen to derive from the composition of the tag set; many can only be corrected through a full syntactic or semantic analysis of the sentence. The paper goes on to consider two very different approaches to the problem of automatically detecting tagging errors. The first uses statistical information that is already at the tagger's disposal; the second attempts to isolate error-prone contexts by formulating linguistic diagnostics in terms of regular expressions over tag sequences. In a small experiment focussing on the preterite/past participle ambiguity, the linguistic technique turns out to be more efficient, while the statistical technique is more effective.

1. Introduction

If taken in isolation, most of the words in any English text – and this may be true for all natural languages – are categorially ambiguous, i.e. they can belong to more than one part of speech (or morpho-syntactic category). This is one of the reasons why almost all machine translation systems incorporate lexical disambiguation rules; this, and the fact that the different categorial realizations of a homograph in one language often receive a distinct translation in other languages. In classical second-generation MT systems, these disambiguation routines are not generally grouped together in a separate module, but are often inextricably entwined with the other rules of the system's parser. In effect, the developers of such systems consider that the higher level syntactic and semantic information required for parsing is also routinely required for the seemingly simpler task of lexical disambiguation. In such systems, moreover, lexical disambiguation is actually a by-product of parsing. That is, for a given sentence, the system rules out incorrect lexical category assignments by failing to achieve a full parse for a path in the input graph that instantiates a particular interpretation of categorially ambiguous words.[1]

In recent years, a number of **specialized** part-of-speech taggers have been developed that take an approach to lexical disambiguation which is radically different from classical G2 MT systems. [Church 1988] describes two of these: his own stochastic part-of-speech tagger, and the part-of-speech program that is a component of Fidditch, Don Hindle's English parser.[2] In contrast to the non-deterministic, full parse approach of the classical MT systems, the taggers that Church describes employ an extremely limited context to assign, in one single pass through the input text, what they calculate to be the most probable part-of-speech to each word. Both programs are designed for very broad coverage and achieve impressively high success rates (see below). The two differ in that Fidditch incorporates linguistic-style disambiguation rules, which are the product of Hindle's ingenuity. (But see [Hindle 1989].) Church's program, on the other hand, assigns part-of-speech tags on the basis of probability estimates that are automatically derived from a large, already tagged training corpus. Two types of probability are used: lexical probability, based on the relative frequency with which a given word has received any particular tag in the training corpus; and contextual probability – the relative frequency with which a given tag has appeared before the following *n* tags in the training corpus. (In general, the value of *n* can vary: in a bigram tagger, the system compiles statistics on the frequencies of 2-tag sequences; in a trigram tagger, the frequencies are of 3-tag sequences. Moreover, the contextual probability of the current tag can be calculated on the basis of the preceding, as well as the following, *n* tags.) The various ways in which these probability estimates can be combined in order to select the most likely part-of-speech for each token is a rather complex question that need not concern us here: see [Church 1988] for a succinct explanation of one option, or chapter 2 of [Foster 1991] for a more detailed account of alternative methods.

Commenting on the 99.5% success rate achieved by his stochastic tagger on the 400 word sample appended to his paper, Church remarks:

> "It is surprising that a local 'bottom-up' approach can perform so well. Most errors are attributable to defects in the lexicon; remarkably few errors are related to the inadequacies of the extremely over-simplified grammar (a trigram model). Apparently, 'long-distance' dependencies are not very important, at least most of the time." (p136-7)

Later in his paper, Church expounds on what he means by "long-distance dependencies", citing examples (repeated in (1a-c) below), which were originally marshalled by Chomsky in the fifties to demonstrate the inadequacy of formalisms like Markov state models as the basis for a theory of grammar.

(1.a) If S1 then S2.
(1.b) Either S3, or S4.
(1.c) The man who said that S5, is arriving today.

The examples in (1a) and (1b) involve correlative conjuncts, while (1c) involves number agreement between the auxiliary *is* and a preceding subject; in both cases, the dependent elements may be separated by sentences of arbitrary length, here symbolized by *S*. Church agrees that such examples of dependencies which potentially extend beyond any fixed length window demonstrate the inadequacy of ngram models for determining grammaticality; his point is that these models may still be acceptable for the task of tagging, because such constructions are relatively infrequent in many varieties of “real-life” texts.

Church's assertion, that long-distance dependencies are not very important for tagging, intrigued me when I first read it. One question it immediately raised was this: Are these the only types of long-distance constructions – or even the most frequent types – that could cause a statistical tagger like his to falter?

2. The Errors

To help me answer this question, I enlisted the assistance of George Foster, a computer scientist working at the CWARC, who recently completed a Master's thesis entitled “Statistical Lexical Disambiguation.” (See [Foster 1991] for the full reference.) In order to explore ways of improving current statistical solutions to the problem of lexical disambiguation, Foster developed an automatic tagging system quite similar in its essentials to that of Church. His taggers were trained on several corpora, both English and French, including a one million word segment of the tagged Lancaster-Oslo-Bergen (or LOB) corpus. To evaluate the effect of varying certain program parameters, Foster tested the system on a disjoint 100,000 word segment of the LOB.[3] When I spoke to him of my interest in the types of errors committed by this kind of tagging program, he responded by providing me with a version of the 100,000 word test segment that made it easy to identify all the errors committed by his tagger, as well as to compile statistics on them.

Table 1 (found at the end of this paper) catalogues the most frequent of these errors, by morpho-syntactic category. The abbreviations are those of the LOB tag set, which is described in detail in [Johannson et al. 1986]; for convenience, I provide a gloss for each tag in the Table. The column headed “correct tag” in Table 1 lists the tags that should have been assigned to the tokens in the test segment, while the figures in the next column give the percentage of the total number of errors attributable to that category. For example, the first entry in the Table states that 10.6% of all corpus errors involved prepositions which Foster's tagger erroneously labelled as something else. The fourth and fifth columns in the Table list the mistaken category assignments by order of frequency; that is, of all the errors involving prepositions, 42.3% were incorrectly tagged as subordinate conjunctions, 16.6% as adverbial particles, and so on.

As shown in lines (1a) and (3a) of the Table, the most frequent error committed by the tagger was to confuse prepositions (IN) and subordinate conjunctions (CS). The two categories are traditionally distinguished by the fact that prepositions take a noun phrase complement, while subordinate conjunctions take a sentential complement. Notice, however, that (in English and in French at least) a number of subordinate conjunctions are identical to prepositions, e.g.: *after*, *before*, *until*, etc. Some grammars of English seek to explain this fact by merging the two categories, and treating these conjunctions **as** prepositions. Prepositions, under this analysis, may subcategorize either an NP or a sentential complement; or, like verbs, they may be intransitive, subcategorizing nothing at all.[4] I cannot defend this analysis in any detail here, referring the interested reader to [Jackendoff 1977] for further justification. Suppose, however, we wanted to adopt it. What would this entail for our statistical tagger?

Firstly, we would have to retag our corpus, replacing the tags for all subordinate conjunctions (CS) with the tag for prepositions (IN)[5]; and then, of course, we would have to retrain the tagger. Neither of these changes would be very difficult to implement. Henceforth, we would no longer require of the tagger that it distinguish between those prepositions that take an NP complement and those that take a sentential complement – something it is not very good at, in any case. In so doing, we would improve the tagger's performance by eliminating its most frequent error. Note that there is nothing mysterious or deceptive about this proposed change to the tag set. What we are proposing would be entirely analogous to suppressing the distinction between verbs that subcategorize an NP and those that subcategorize an S. If the LOB doesn't distinguish such tags for verbs, why should it do so for prepositions?

Generally speaking, a given tag set may be more or less suitable for certain applications, and in principle, there is no reason to consider the one used in the LOB corpus (which itself is a refinement of the tag set used in the Brown University corpus) as absolutely inviolate. The tagging programs themselves, of course, are independent of any particular tag set.[6] On the other hand, there may be strong practical reasons that militate against the introduction of wholesale changes to the tag set, reasons that have to do with the enormous human effort required to produce a large-scale, correctly tagged reference corpus like the LOB or the Brown. In this particular case, such objections do not seem to apply. In fact, the proposed change does not necessarily eradicate the distinction between prepositions and subordinate conjunctions. Within a full-blown NLP system, it would still be possible to read this distinction off the lexical subcategorization frames associated with the various sub-classes of prepositions.

Another category distinction on which the tagger frequently falters is that between past participles (VBN) and preterites (VBD): cf lines (6a) and (7a) in Table 1. For all regular verbs of English, and many irregulars as well, these forms are identical, both being marked by the "-ed" suffix. From the point of view of a strictly local, statistical tagger, this too could be considered a long-distance dependency, though not exactly of the same sort as was illustrated by the examples in (1a-c) above. There, the choice of the second correlative conjunct depends on the form of the first; and similarly, the form of the auxiliary verb depends on the person and number of the preceding subject. Here, the choice of the past participle (VBN) tag may depend on the presence of a preceding perfect or passive auxiliary; but it may also depend on the "-ed" form occurring alone in a subjectless non-finite clause. Similarly, the correct assignment of the preterite (VBD) tag depends on the absence of other auxiliaries in a finite clause. In these cases, what is crucial is not so much the distance, or number of words separating the "-ed" form from some governing element; rather, it is the nature or function of the clause in which the “-ed” form appears; and this, as any traditional linguist will tell you, can only be determined by means of a full parse of the entire sentence. (The term “global dependency” might be preferable to long-distance dependency for cases like these.) Insofar as a tagger is concerned, the real question, of course, is to what extent these functional distinctions are reflected in local dependencies that show up in a statistically significant manner.

The famous garden path example, repeated in (2.a) below, can serve to illustrate the problem.

(2.a) The horse raced past the barn fell

The first time we read this sentence, we cannot help but take *raced* as a preterite; only when we come upon *fell* in sentence-final position do we realize that we've been led down the garden path, and so must revise our analysis of the initial “-ed” form – something the tagger cannot do. [Church 1988] maintains that problematic cases like these are the exception rather than the rule, and do not jeopardize the general feasibility of automatic tagging; and in this he may be right. On the other hand, our data show that the contextual clues which would allow a trigram tagger to correctly distinguish past participles from preterites on the basis of strictly local information are absent in a significant proportion of cases. Consider in this regard the examples in (2.b-e) below, which are taken from Foster's tagging of the LOB. (*VBD here marks a past participle that the tagger incorrectly labelled as a preterite, and *VBN, a preterite incorrectly tagged as a participle.)

(2.b) the excuse now put/*VBD forward that ...
(2.c) I have heard it said/*VBD that ...
(2.d) share and deposit balances increased/*VBN to ...
(2.e) the fifth boy, Sidney, left/*VBN for the navy ...

Of all the preterites that appear in the 100,000 word test corpus, 6.9% were incorrectly tagged, as were 4.5% of all past participles. There is little reason to believe, moreover, that these error rates would drop significantly, were the tagger's window expanded to include four or five tokens.

The last class of errors I want to consider arises from the tagger's confusion of nouns (NN) and adjectives (JJ): cf lines (2a) and (5a) in Table 1. There are two sources to the problem here: first, the fact that adjectives and nouns have similar distributions, as in pre-head noun position and after *be*; and second, the fact that many typical adjectives can appear in basic nominal positions. The examples in (3a-c) and (4a-b) below serve to illustrate: (Here, the tags following the slash designate the correct category assignments.)

(3a) a dull orange/JJ shirt
lemon and orange/NN groves

(3b) a reduction of working/NN hours
a collection of working/JJ women

(3c) individual/NN and group piece-work
individual/JJ or collective decisions

(4a) I hope not to be pointing out the obvious/JJ if ...

(4b) It was no good/JJ, and they both knew it.

Even for the human revisers of the LOB corpus, examples like these are often difficult to resolve, and [Johansson et al. 1986] invoke all sorts of criteria, including various types of paraphrase and a word's other distributional possibilities, to justify their selection of JJ or NN. In the end, they conclude that “no claim can be made for complete consistency in the distinction between adjective and noun in attributive position” (p64), and the same could be said for the tagging of adjectives in typical nominal position. These, then, are among the most difficult cases of categorial homography for a statistical tagger to resolve, where often, as Church says, “no amount of syntactic parsing will help” (p137). To see why, we only need reconsider the examples in (3a). It is not sufficient here to know that shirts can be orange coloured, and that oranges grow in groves of trees; to rule out the **incorrect** readings, one also needs to know that it is highly improbable for shirts to be made of oranges, or for groves to be orange coloured. The correct category assignment, in other words, does not depend on some other element in the sentence, which may be arbitrarily far removed; rather, it depends on detailed semantic and seemingly arbitrary pragmatic knowledge of the sort that has long been the bane of large-scale NLP applications.

One of the appeals of the new corpus-based approaches to NLP is the belief that they will somehow allow us to avoid having to formalize and program vast amounts of such knowledge in order to improve system performance. On its own, however, a statistical tagger would appear inherently incapable of dealing with cases like these. As we've seen, the only knowledge the tagger has at its disposal is how often the word *orange* appears as a noun in some training corpus, and how often it appears as an adjective. It also knows from its contextual statistics the relative frequencies of adjectives and nouns in various attributive positions. But for examples like those in (3a-c), the only consistently correct way to arrive at the permissible or impossible semantic relations which determine the appropriate category assignments is to examine the actual words with which *orange* occurs in context. Statistical taggers of the sort being considered here do not do this. Hence, in such cases, and to the extent that noun/adjective frequencies are distributed fairly evenly, the tagger is shooting in the dark; it is doubtful that it can do much better than chance.[7]

3. Automatic Error Detection

Once one understands how a statistical ngram tagger operates, and in particular the limitations of the contextual constraints which it allows for, there is nothing very surprising about the error results described above.[8] Moreover, there is an obvious objection to the sort of error analysis we have conducted, and it is this: Why bother? When a tagger achieves an overall success rate of 98%, is it not mere niggling to dissect its occasional failings?

There are two ways to respond to this objection. The first is to place these success rate figures, which may be somewhat misleading, in clearer perspective. When Church claims a program performance of 95-98% correct, he is including all the tokens in his test corpus, a large proportion of which are unambiguous to start with.[9] When these figures are readjusted, taking into account only ambiguous tokens, the success rate drops to around 90%. But even this appears deceptively high. Viewed another way, a 90% success rate is equivalent to one tagging error every ten ambiguous words – or roughly one error every two lines on a page like this one.

The second response to the "Why bother?" objection starts from this more sober assessment of tagger performance and confronts it with situations where it is simply not acceptable. Suppose, for example, we were developing a large, new training corpus for a statistical tagging system, and wanted to use an existing tagger to assign an initial tagging to the corpus. Or suppose we were developing an NLP application which would include a tagger as part of its front end analysis component – say, a machine translation system, where the tagger would pre-process the input text, so that the parser would receive a labelled string of words that was categorially disambiguated. In both cases, there can be little tolerance for tagging errors; this is obvious in the case of the training corpus, and it is confirmed for the MT application by [Brown et al. 1992].[10] Either such tagging errors will have to be corrected, or they will markedly degrade the performance of the system fed by the tagger. An important question that arises in this context, therefore, is whether one can **automatically** identify those contexts where the tagger tends to falter. Following [Foster 1991], we will refer to this as the tagging error detection problem.

One interesting possibility is that error-prone contexts, like those we examined in Section 2 above, can be automatically detected using information that is already at the tagger's disposal. This idea of statistical error detection is explored in some detail in chapter 6 of [Foster 1991]. Foster's basic hypothesis is that "tagging errors correspond to situations in which the model assigns similar probabilities to competing alternatives" (p75). A statistical tagger, recall, calculates probabilities for all the possible tags of a given token. In the clearest cases, the probability score assigned to the correct tag will be significantly greater than the scores of the competing tags. Foster's hypothesis is just the corollary of this, i.e. that tagging errors tend to occur in situations where the two best tags on a given token have scores that are relatively close, indicating a degree of system uncertainty, as it were. Just how close these scores have to be is a parameter that can be modulated: if the required difference between the two scores is set at some minimum value, the detection program will flag a relatively small number of tokens; on the other hand, if the difference between the two scores is increased, more tokens will be flagged – including, presumably, a larger number of errors. There is a clear trade-off here, and various external factors may determine the most appropriate setting on the continuum for a given application.

One such external factor is how, or by whom, these error flags will subsequently be processed. If the tagger is being used to feed a parser, one possibility would be to build a description of these error-prone configurations into the tagging program itself, in a way that would block or suspend the selection of a tag in certain of these situations. In the case of the preterite/ past participle ambiguity, for example, and in the absence of positive contextual clues like the passive or perfect auxiliary, it may be preferable for the tagger NOT to hazard a guess based on insufficient context. Rather, the tagger could pass the ambiguity on to the parser, where it hopefully would be resolved in the course of a full sentential analysis.[11] The objection to this strategy, of course, is that it sacrifices the large proportion of correctly tagged "-ed" forms in order to avoid mis-tagging a much smaller proportion of errors. (And where there is no parser downstream, as in the development of a training corpus, it is simply not applicable.) On the other hand, if the potential errors are to be reviewed by a human, one may want to ease the burden of revision by flagging proportionally fewer tokens, and allowing more errors to go undetected. As an illustration of the trade-off, Foster shows that his statistical algorithm will detect all but 1% of the tagging errors in a corpus if the difference between the two best scores is set in such a way that 15% of all tokens are flagged.[12]

An altogether different approach to the detection problem would be to tap the linguistic knowledge gleaned from our error analysis in an effort to identify just those configurations where the tagger tends to falter most frequently. One method would be to formulate descriptions of these error-prone contexts in terms of regular expressions over tag sequences. Applied to the tagger's output, these linguistic (or symbolic) diagnostics would flag potential errors, just as the statistical detection algorithm does. Formulated as regular expressions, they would have the advantage of being able to examine a much wider and more complex context than the tagger, while remaining within the same formal complexity class, i.e. their pattern matching can be done in time which is linear in the length of the input.

To test this approach, I decided to mount a small experiment, focussing on the preterite/ past participle ambiguity, a problem which, as mentioned, is at least amenable to syntactic resolution. Having extracted from G. Foster's 100,000 word test corpus all the cases where his tagger mistook VBN for VBD, and vice versa, I formulated a set of symbolic diagnostics, expressed in terms of Boolean conditions on strings of tags. These can be roughly paraphrased as follows: (Recall that VBD is the LOB tag for preterites, and VBN, for past participles.)

(5a) In a sentence where a VBD immediately follows a co-ordinate conjunction, flag the VBD if the string preceding the conjunction contains a passive or perfect auxiliary and an adjacent VBN.

(5b) In a sentence where a VBN immediately follows a co-ordinate conjunction, flag the VBN if the string preceding the conjunction contains a VBD.

(6) Flag a VBN if it is the sole verbal form between two full-stop punctuation marks, or between a relative pronoun and a full-stop punctuation.

(7) Flag a VBD if it is separated from a preceding passive or perfect auxiliary by one or more adverbs, or by a personal pronoun.

Taken together, (5a) and (5b) are meant to express the tendency in conjoined clauses, where a "higher" subject and auxiliary are not repeated in a "lower" conjoined clause, for past participles to be conjoined with past participles and preterites to be conjoined with preterites. I stress that this is a (perhaps stylistic) tendency, and not a hard and fast grammatical rule. As stated, (5a-b) would result in the flagging – signalled here by the introduction of a caret – of the incorrectly tagged VBD in (8a) and the incorrect VBN in (8b).

(8a) Those are often left/VBN in their pots and laid/^VBD on their sides ...

(8b) The RAF men supervised/VBD my placing of them and apparently approved/^VBN.

(6) is intended to capture the intuition that if an "-ed" form is the sole verb in a sentence, i.e. if there are no other auxiliaries or verbal forms, it is unlikely to be a past participle. One problem with this, of course, is that not every unit that occurs between full-stop punctuation in a corpus is a sentence; headings and titles may not be formally distinguished, or marked up differently from sentences, and they may include a VBN. Still, (6) will result in the flagging of incorrectly tagged participles in examples like (9a-b).

(9a) Reserves increased/^VBN by 673,000 to 2,930,000.

(9b) ... showing how/WRB the different groups performed/^VBN on the various experiments.

Finally, (7) is meant to flag mis-tagged past participles like those in (10a-b).

(10a) The United States and Latin America have also/RB recently/RB tightened/^VBD up their immigration controls.

(10b) ... or, alternatively, is it/PP3 merely utilized /^VBD to bring about a general reduction

The linguistic diagnostics in (5-7) were translated into regular expressions and applied to the tagged test corpus. Before presenting the numerical results of the experiment, I should point out that these diagnostics aim for a certain degree of precision. Based on a linguistic analysis of frequent tagging errors, they are intended to identify just the contexts in which those errors occur, without flagging too many correctly tagged "-ed" forms. In other words, there is a trade-off here too, between diagnostics that are **efficient** in not flagging too many false errors, and diagnostics that are **effective** in not missing too many true errors. Limited to patterns and conditions on strings of tags, however,I was not able to formulate linguistic descriptions that could capture all, or even most of the VBN/VBD error configurations without engendering a high level of noise. As mentioned above, the only consistently sure way to disambiguate a preterite from a past participle is via a full parse of the entire sentence. These symbolic diagnostics, while less expressively restricted than a trigram tagger, are still a far cry from having the full combinatorial power of a non-deterministic parser. What one would hope is that they would still allow us to identify an interesting proportion of the errors committed by the tagger, in such a way that could perhaps reduce the burden on a human reviser, without requiring all the computational power of a parser.

So then, how did the diagnostics in (5-7) fare? Collectively, they resulted in 72 flags being raised on ambiguous "-ed" forms;[13] of these, 49 corresponded to incorrectly tagged VBN or VBD, for an efficiency rate of 68%. The effectiveness of these diagnostics, on the other hand, was substantially lower. In all, the test corpus contained 166 incorrectly tagged preterites and past participles; the diagnostics in (5-7) flagged 49 of these, for an effectiveness rate of 30%. How does this performance compare with Foster's statistical flagging algorithm, described above? For such a comparison to be fair, the statistical detector should raise about the same number of error flags as the symbolic detection method. When this was done, it turned out that 35 of the 72 flags raised by the statistical detector corresponded to incorrectly tagged VBN or VBD, for an efficiency rate of 49%.

4. Conclusion

What can we reasonably conclude from these results? In headier moments, we might be tempted to consider this small experiment in error detection to be a reflection of the more general methodological issues that are the focus of this Conference, and to project these results onto a higher theoretical plane. This would be somewhat hasty, however, and probably ill-advised. For one thing, we do not yet know to what extent the particular problem we have selected – i.e. the preterite/past participle ambiguity – is representative of the possibilities of symbolic error detection in general. (Nor do we know, for that matter, whether our diagnostics for this problem would yield the same results if applied to another tagged corpus.) What's more, it is important to recall that though we have contrasted two error detection techniques, one symbolic and the other statistical, both operate on an already tagged corpus, produced as it happens by a stochastic tagger. All that we can conclude at this point, then, is that the difference in efficiency rates between the statistical and symbolic detection methods is significant enough to merit pursuing our investigation. If we manage to formulate linguistic diagnostics for other frequent tagging errors, and if these yield results comparable to those reported here, then we could perhaps envisage a mode of cooperation between the statistical and the linguistic approaches, whereby the great bulk of the work of tagging is done by stochastic methods, and the fine tuning (or the error detection) is done with the help of linguistic techniques like those we have presented here.[14]

Acknowledgements

This paper is based on George Foster's excellent Master's thesis. I am indebted to him, both for his explanations of points in the thesis and for kindly providing me with the supplementary data on error frequencies. All responsibility for errors of interpretation is mine alone. Pierre Isabelle, Michel Simard, Marc Dymetman and Marie-Louise Hannan all provided comments on an earlier version of this paper, for which I also express my gratitude.

Notes

1. One second-generation MT system that was entirely typical of this approach was TAUM-Aviation. See [Isabelle 1987] for a full description of this system and a more detailed discussion of lexical disambiguation within a classical G2 framework.

2. A concise survey of other statistical taggers may be found in Chapter 2 of [Foster 1991].

3. The total LOB corpus is one million words long. Foster's one million word training segment included punctuation; this is why he had 100,000 words left over for testing.

4. Intransitive prepositions correspond to those non "-ly" forms that are traditionally classed as adverbs, though their distribution is much closer to that of prepositions than to other adverb classes. A form like *upstairs*, for example, can occur in the complement to a subject noun phrase, or alone after the copula – positions where "-ly" adverbs do not generally appear. These are the so-called adverbs that the tagger frequently confuses with prepositions, as shown in lines (4.a) and(1.b-c) of the Table.

5. Alternatively, we could retain CS for just those prepositions, like *because*, that only subcategorize an S. This would have the advantage of not weakening the tagger's contextual statistics, for example on the distribution of sentential adverbs that appear after CS though not after IN.

6. In fact, [Foster 1991] employs a reduced version (or subset) of the full LOB tag set, alongside the full set.

7. Even statistical descriptions based on the notion of word association like [Church and Hanks 1990], which explicitly seek to elicit from huge corpora the most salient co-occurrences of a particular word in context, would seem to be helpless here. Church and Hanks claim that their methodology can provide "disambiguation cues for parsing highly ambiguous syntactic structures such as noun compounds, conjunctions and prepositional phrases" (p22); but nowhere in their paper do they discuss examples like (3a), where it is not the internal grouping of the noun phrase that is in question, but rather its labelling.

8. We have focussed on the most frequent types of errors committed by the tagger; but if we consider another type of long-distance dependency – say, the ambiguity of *that* as a subordinate conjunction versus a relative pronoun – here too we could predict a priori that the tagger would not fare too well. Why? Because in many cases, the only sure way to identify the relative pronoun is to locate a gap in a following clause; and these notions of gap and clause are simply not available to a trigram tagger. As it turns out, close to 20% of the occurrences of *that* as relative pronoun in the test corpus are mis-tagged as subordinate conjunctions.

9. By way of example, approximately 45% of the tokens in the LOB corpus are categorially unambiguous, according to [Foster 1991].

10. The authors cite errors in grammatical tagging as one of the reasons why their English question inversion transformation, which is intended to reduce local statistical variety, succeeds only about 40% of the time.

11. This is not to suggest that the parser will necessarily be able to resolve **all** the most difficult cases of categorial ambiguity, such as the NN/JJ ambiguity discussed in Section 2 above. But given the appropriate semantic and pragmatic information, a robust parser at least has a chance; while the tagger, for the cases that are inherently beyond its ken, can do no better than chance.

12. At this setting, having a human reviser check for all the potential errors in a 100,000 word corpus could be quite a daunting task; for it would involve more than just verifying the tags on 15,000 tokens. In order to make a decision on a given tag, the reviser will generally have to read a good portion of the sentence in which it occurs. If he checks 6-7 words around each of the flagged tokens, he will have read through the entire corpus.

13. In addition to the patterns described in (5-7), there was also a minor test that flagged the second of two contiguous VBN.

14. Notice, incidentally, that the effort invested by a linguist in elaborating and testing these sorts of diagnostics is partially amortized each time the diagnostics are applied to a new corpus. On the other hand, the time invested by a reviser in correcting tagging errors is not recoverable on the next corpus.

References

Brown, Peter et al. (1992), “Analysis, Statistical Transfer, and Synthesis in Machine Translation”, paper submitted to the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, Montreal, Canada, June 1992.

Church, Kenneth Ward, (1988), “A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text”, in the *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, February 1988, Association for Computational Linguistics.

Church, Kenneth Ward, and Patrick Hanks, (1990), “Word Association Norms, Mutual Information, and Lexicography”, in *Computational Linguistics*, Vol.16 No.1, March 1990.

Foster, George F., (1991), “Statistical Lexical Disambiguation”, Master's thesis, McGill University, School of Computer Science, Montreal, Canada.

Hindle, Donald, (1989), “Acquiring Disambiguation Rules from Text,” in *Proceedings of the 27th Annual ACL Meeting*, Vancouver, Canada, June 1989, Association for Computational Linguistics.

Isabelle, Pierre, (1987), “Machine Translation at the TAUM Group”, in *Machine Translation Today: The State of the Art*, Margaret King (ed.), Edinburgh University Press, Edinburgh.

Jackendoff, Ray, (1977), *X-Bar Syntax: A Study of Phrase Structure*, MIT Press, Cambridge, Mass.

Johannson, Stig et al. (1986), *The Tagged LOB Corpus: Users' Manual*, Norwegian Computing Centre for the Humanities, Bergen.

Table 1:

line	correct tag	%	incorrect tag	%
(1. a)	IN (preposition)	10.6	CS (subordinate conj)	42.3
.b)			RP (adverbial particle)	16.6
.c)			RB (adverb)	10.4
(2. a)	NN (sing. common noun)	10.2	JJ (adjective)	39.7
.b)			VBG (verb~ing)	20.3
.c)			VB (verb~base)	14.2
(3. a)	CS (subordinate conj)	9.8	IN (preposition)	55.2
.b)			DT (sing determiner)	13.9
.c)			QL (qualifier)	9.9
(4. a)	RB (adverb)	8.6	IN (preposition)	31.6
.b)			JJ (adjective)	15.3
.c)			AP (post-det)	13.8
.d)			CS (subordinate conj)	9.2
(5. a)	JJ (adjective)	5.3	NN (sing. common noun)	31.4
.b)			VBN (past participle)	20.7
.c)			RB (adverb)	19.8
(6. a)	VBN (past particple)	5.2	VBD (preterite)	67.2
.b)			JJ (adjective)	28.6
(7. a)	VBD (preterite)	4.6	VBN (past participle)	91.3
(11.a)	RP (adverbial particle)	2.9	IN (preposition)	70.8
.b)			RB (adverb)	18.5
(14.a)	NNS (plural noun)	2.0	NN (sing. common noun)	60.9