Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?

Christopher D. Manning

Departments of Linguistics and Computer Science
Stanford University
353 Serra Mall, Stanford CA 94305-9010
manning@stanford.edu

Abstract. I examine what would be necessary to move part-of-speech tagging performance from its current level of about 97.3% token accuracy (56% sentence accuracy) to close to 100% accuracy. I suggest that it must still be possible to greatly increase tagging performance and examine some useful improvements that have recently been made to the Stanford Part-of-Speech Tagger. However, an error analysis of some of the remaining errors suggests that there is limited further mileage to be had either from better machine learning or better features in a discriminative sequence classifier. The prospects for further gains from semi-supervised learning also seem quite limited. Rather, I suggest and begin to demonstrate that the largest opportunity for further progress comes from improving the taxonomic basis of the linguistic resources from which taggers are trained. That is, from improved descriptive linguistics. However, I conclude by suggesting that there are also limits to this process. The status of some words may not be able to be adequately captured by assigning them to one of a small number of categories. While conventions can be used in such cases to improve tagging consistency, they lack a strong linguistic basis.

1 Isn't Part-of-Speech Tagging a Solved Task?

At first glance, current part-of-speech taggers work rapidly and reliably, with per-token accuracies of slightly over 97% [1–4]. Looked at more carefully, the story is not quite so rosy. This evaluation measure is easy both because it is measured per-token and because you get points for every punctuation mark and other tokens that are not ambiguous. It is perhaps more realistic to look at the rate of getting whole sentences right, since a single bad mistake in a sentence can greatly throw off the usefulness of a tagger to downstream tasks such as dependency parsing. Current good taggers have sentence accuracies around 55–57%, which is a much more modest score. Accuracies also drop markedly when there are differences in topic, epoch, or writing style between the training and operational data.

Still, the perception has been that same-epoch-and-domain part-of-speech tagging is a solved problem, and its accuracy cannot really be pushed higher. I

think it is a common shared meme in at least the U.S. computational linguistics community that interannotator agreement or the limit of human consistency on part-of-speech tagging is 97%. As various authors have noted, e.g., [5], the second wave of machine learning part-of-speech taggers, which began with the work of Collins [6] and includes the other taggers cited above, routinely deliver accuracies a little above this level of 97%, when tagging material from the same source and epoch on which they were trained. This has been achieved by good modern discriminative machine learning methods, coupled with careful tuning of the feature set and sometimes classifier combination or semi-supervised learning methods. Viewed by this standard, these taggers now *clearly exceed* human performance on the task. Justifiably, considerable attention has moved to other concerns, such as getting part-of speech (POS) taggers to work well in more informal domains, in adaptation scenarios, and within reasonable speed and memory limits.

What is the source of the belief that 97% is the limit of human consistency for part-of-speech tagging? It is easy to test for human tagging reliability: one just makes multiple measurements and sees how consistent the results are. I believe the value comes from the `README.pos` file in the `tagged` directory of early releases of the Penn Treebank. It suggests that the "estimated error rate for the POS tags is about 3%".[1] If one delves deeper, it seems like this 97% agreement number could actually be on the high side. In the journal article on the Penn Treebank [7], there is considerable detail about annotation, and in particular there is description of an early experiment on human POS tag annotation of parts of the Brown Corpus. Here it was found that if two annotators tagged for POS, the interannotator disagreement rate was actually 7.2%. If this was changed to a task of correcting the output of an automatic tagger (as was done for the actual Penn Treebank), then the disagreement rate dropped to 4.1%, and to 3.5% once one difficult text is excluded. Some of the agreement is then presumably both humans adopting the conventions of the automatic POS tagger rather than true human agreement, a topic to which I return later.

If this is the best that humans can give us, the performance of taggers is clearly at or above its limit. But this seems surprising – anyone who has looked for a while at tagger output knows that while taggers are quite good, they regularly make egregious errors. Similarly, examining portions of the Penn Treebank by hand, it is just very obvious that there are lots of errors that are just mistakes rather than representing uncertainties or difficulties in the task. Table 1 shows a few tagging errors from the beginning of section 02 of the training data.[2] These are all cases where I think there is no doubt about what the correct tag should be, but that nevertheless the annotator failed to assign it. It seems

[1] This text appears up through LDC95T7 Treebank release 2; the statement no longer appears in the much shorter README included in the current LDC99T42 Treebank release 3). This error rate is also mentioned in [7, pp. 327–8].

[2] My informal impression is that the accuracy of sections 00 and 01 is considerably worse, perhaps reflecting a "burn in" process on the part of the annotators. I think it is in part for this reason that parsers have been conventionally trained on sections 02–21 of the Penn Treebank. But for POS tagging, most work has adopted the splits introduced by [6], which include sections 00 and 01 in the training data.

clear that the inter-annotator agreement of humans depends on many factors, including their aptitude for the task, how much they are paying attention, how much guidance they are given and how much of the guidance they are able to remember. Indeed, Marcus et al. [7, p. 328] express the hope that the POS error rate can be reduced to 1% by getting corrections from multiple annotators, adjudicating disagreements, and using a specially retrained tagger. However, unfortunately, this work never took place. But using the tools developed over the last two decades given the existence of the Penn Treebank, we are now in a much better position to do this, using semi-automated methods, as I discuss below.

Table 1. Examples of errors in Penn Treebank assigned parts-of-speech, from section 02 of the *WSJ*.

Time , the/DT largest/JJ ***newsweekly/RB*** *, had average circulation of*
Correct: **newsweekly/NN**
below the $ 2.29 billion value United Illuminating ***places/NNS*** *on its bid*
Correct: **places/VBZ**
Rowe also noted that political concerns also ***worried/VBN*** *New England Electric .*
Correct: **worried/VBD**
Commonwealth Edison now faces an additional court-ordered refund on its summer/winter rate differential collections ***that/IN*** *the Illinois Appellate Court has estimated at $ 140 million .*
Correct: **that/WDT**
Joseph/NNP M./NNP Blanchard/NNP , 37 , vice president , engineering ; Malcolm/NNP ***A./NN*** *Hammerton/NNP*
Correct: **A./NNP**

2 Approaching the asymptote: Continuing to push up POS tagging numbers

Since the time of our last POS tagger paper [1], I've added a few features that have slightly pushed up the performance of the Stanford POS tagger. I give the details of those models here. But it is noticeable that they do not improve overall performance by very much. Other people seem to be hitting the same wall, and while there are fractionally better results from others, none are much better. Suppose somehow that more machine learning magic can get numbers up from 97.3% per-token accuracy to 97.5% per-token accuracy. That would still mean that the last decade will only have solved about 1/6 of the errors remaining in part of speech taggers.

2.1 Incremental improvements

The experiments I present here describe incremental work: There are no big changes to the architecture, but some improvements in the features, parameters,

and learning methods give small incremental gains in POS tagging performance, bringing it close to parity with the best published POS tagging numbers in 2010. These numbers are on the now fairly standard splits of the *Wall Street Journal* portion of the Penn Treebank for POS tagging, following [6].[3] The details of the corpus appear in Table 2 and comparative results appear in Table 3.

Table 2. *WSJ* corpus for POS tagging experiments.

Set	Sections	Sentences	Tokens	Unknown
Training	0-18	38,219	912,344	0
Development	19-21	5,527	131,768	4,467
Test	22-24	5,462	129,654	3,649

Table 3. Tagging accuracies with different feature templates and other changes on the *WSJ* 19-21 development set.

Model	Feature Templates	# Feats	Sent. Acc.	Token Acc.	Unk. Acc.
3GRAMMEMM	See text	248,798	52.07%	96.92%	88.99%
NAACL 2003	See text and [1]	460,552	55.31%	97.15%	88.61%
Replication	See text and [1]	460,551	55.62%	97.18%	88.92%
Replication′	+rareFeatureThresh = 5	482,364	55.67%	97.19%	88.96%
5W	$+\langle t_0, w_{-2}\rangle, \langle t_0, w_2\rangle$	730,178	56.23%	97.20%	89.03%
5WSHAPES	$+\langle t_0, s_{-1}\rangle, \langle t_0, s_0\rangle, \langle t_0, s_{+1}\rangle$	731,661	56.52%	97.25%	89.81%
5WSHAPESDS	+ distributional similarity	737,955	56.79%	97.28%	90.46%

3GRAMMEMM shows the performance of a straightforward, fast, discriminative sequence model tagger. It uses the templates $\langle t_0, w_{-1}\rangle$, $\langle t_0, w_0\rangle$, $\langle t_0, w_{+1}\rangle$, $\langle t_0, t_{-1}\rangle$, $\langle t_0, t_{-2}, t_{-1}\rangle$ and the unknown word features from [1]. The higher performance NAACL 2003 tagger numbers come from use of a bidirectional cyclic dependency network tagger, which adds the feature templates $\langle t_0, t_{+1}\rangle$, $\langle t_0, t_{+1}, t_{+2}\rangle$, $\langle t_0, t_{-1}, t_{+1}\rangle$, $\langle t_0, t_{-1}, w_0\rangle$, $\langle t_0, t_{+1}, w_0\rangle$, $\langle t_0, w_{-1}, w_0\rangle$, $\langle t_0, w_0, w_{+1}\rangle$ The next line shows results from an attempt to replicate those numbers in 2010. The results are similar but a fraction better.[4] The line after that shows that the numbers are pushed up a little by lowering the support threshold for including rare word features to 5. Thereafter, performance is improved a little by adding features. 5W adds the words two to the left and right as features, and 5WSHAPES also adds word shape features that we have described for named en-

[3] In this paper, when I refer to "the Penn Treebank", I am actually referring to just the *WSJ* portion of the treebank, and am using the LDC99T42 Treebank release 3 version.

[4] I think the improvements are due to a few bug fixes by Michel Galley. Thanks!

tity recognition elsewhere [8].[5] These features map words to equivalence classes based on character type, such as *Mexico* to *Xxxx* and *IA-64* to *XX-dd*. Some of the other recent taggers cited earlier have made use of even more higher order features and conjunctive features [2, 10], but in our tagger they seem to provide marginal to negative gains. Something that does really help is adding features for words based on induced distributional similarity classes, as shown on the last line.[6] Here, we use the method and code of [11], though other methods for introducing distributional similarity classes would probably work roughly as well. While the overall gains on the *WSJ* are modest, taken together, these features and the word shape features give a significant gain in performance on unknown words: errors on unknown words are reduced by 13% (relative). And one would expect these features to be even more useful when the tagger is subsequently used on text from other domains or epochs. Note, however, that the last line is for a model where the distributional similarity classes were trained separately on about 300 million words of data in an unsupervised fashion, whereas all the other models are trained only on the *WSJ* training set.

I present these numbers to show that while small amounts of progress remain possible, we clearly seem to be entering an era of diminishing returns. It seems like about 2.4% of the remaining 2.6% error rate might need to be approached from a different angle.[7]

2.2 Splitting tags

I have shown in other work that parsing performance on the Penn Treebank can be improved enormously by splitting certain of the categories, both part-of-speech and phrasal categories, and parsing with the resulting split-category treebank grammar [12]. One might reasonably think that the same strategy could be applied successfully to the POS tagging problem, especially as a number of the most useful state splits for parsing are splits of part-of-speech categories. But, unfortunately, splitting tags seems to be largely a waste of time for the goal of improving POS tagging numbers. A thorough exploration of the possibilities can be found in [13]. My own more limited experimentation points in the same direction.

3 Error Analysis

How, then, can we solve the other 5/6 of the errors of POS taggers? An examination of the things that taggers get wrong on same-domain test text makes it

[5] As far as I am aware, features of this sort were first introduced by Collins [9].

[6] This line corresponds to the released version 3.0 of the Stanford POS Tagger, available at *http://nlp.stanford.edu/software/tagger.shtml*

[7] Since this investigation was part of a series of experiments on different models, they were all evaluated only on the development set (section 19–21). It is now clear from several studies, including the numbers below, that the final test set is a bit easier, and it could be expected that final test set numbers would be almost 0.1% higher. See table 6 for results on the final test set.

clear that little of the remaining error is going to be solved by better local features of the kind used by current state-of-the-art sequence model taggers. How are we going to get the rest? To answer that, we need to understand what kinds of errors there are. In Table 4, I give a rough breakdown of where we need to look.[8]

I did a small error analysis, taking a sample of 100 errors from section 19 of the treebank. I divided errors into seven classes, as shown in Table 4. Many errors are hard to classify. When things were unclear, I allowed an error to be assigned to two classes, giving it 1/2 a point under each. I exemplify the seven classes below.

Table 4. Frequency of different POS tagging error types.

Class	Frequency
1. Lexicon gap	4.5%
2. Unknown word	4.5%
3. Could plausibly get right	16.0%
4. Difficult linguistics	19.5%
5. Underspecified/unclear	12.0%
6. Inconsistent/no standard	28.0%
7. Gold standard wrong	15.5%

1. Lexicon gap: Here, the word occurred a number of times in the training data, but never with the tag which it has in this context. Given the nature of discriminative POS taggers, it is always going to be very difficult for context to override lexical features in this situation. For example, below, *slash* is clearly a noun, but in the training set, it occurs only but several times as a verb.

a/DT 60/CD %/NN slash/NN in/IN the common stock dividend

2. Unknown word: Here the tagger has to rely only on context features, and contexts are often ambiguous. For example, below, *substandard* is a word which does not appear in the training data and it is also very reasonable for a POS tagger to guess that it might be a noun (as it did).

blaming the disaster on/IN substandard/JJ construction/NN

3. Could plausibly get right: Here, you could imagine a sequence model tagger with a context of a few words or tags on either side getting the right answer, though it may be quite difficult in practice. For example, below, it seems like a sequence tagger should be able to work out that *overnight* is here functioning as an adverb rather than an adjective (the tag it chose), since it is here a verb modifier not pre-modifying a noun.

market/NN players/NNS overnight/RB in/IN Tokyo/NNP began bidding up oil prices

[8] An early, but somewhat imprecise, discussion of the different sources of tagging disagreement can be found in [14].

4. **Difficult linguistics: Needs much syntax/semantics/discourse:** Here, it seems very clear that determining the right tag requires broad contextual knowledge that must be beyond a sequence tagger with local features. For example, below, a tagger just cannot correctly choose between the present (VBP) and past (VBD) tag for *set* without an understanding of a multi-sentence discourse context, and happens to choose wrongly.

   They/PRP set/VBP up/RP absurd/JJ situations/NNS , detached from reality

5. **Underspecified/unclear:** The tag is underspecified, ambiguous, or unclear in the context. There are several common cases of this, such as whether to choose a verbal or adjectival tag for words which have a participial inflectional form and modify a head, and whether to choose a verbal or noun tag for gerunds. While there are linguistic tests that can be used to distinguish the two categories in both these cases, often in particular contexts the correct analysis is just underspecified. For example, below, it is unclear whether *discontinued* should be regarded as an adjective or verbal participle.

   it will take a $ 10 million fourth-quarter charge against/IN discontinued/JJ operations/NNS

6. **Gold standard inconsistent or lacks guidance:** Here, there should be a right answer, but the tagging manual does not define what to do and in practice the annotators have been inconsistent, so it is not surprising that the tagger gets such things right only half the time by chance. For example, for expressions like *the '30s* below, or indeed corresponding ones like *the 1930s*, the treebank is inconsistent in sometimes tagging them as CD and at other times as NNS. There should be a clear answer here which should be consistently used, but none was defined, and human annotators were inconsistent. (If the tag CD is construed fairly strictly as cardinal numbers – for example, ordinals are definitely excluded and tagged as adjectives (JJ), then it seems to me like these expressions shouldn't be tagged CD, and that NNS is correct, and below we retag in this fashion, but in the Treebank, the two taggings are almost exactly equally common, with a couple of tokens also tagged as NN, to add variety.)

   Orson Welles 's Mercury Theater in/IN the/DT '30s/NNS ./.

7. **Gold standard wrong:** The tag given in the gold standard is clearly wrong. For example, below, the tag of VB for *hit* is just wrong. It should be a VBN, as the passive participle complement to *got*. Other examples of this sort appeared in Table 1.

   Our market got/VBD hit/VB a/DT lot/NN harder/RBR on Monday than the listed market

What conclusions can we draw? While semi-supervised methods like the distributional similarity classes above are very useful for handling unknown words, their ability to improve overall tagger performance numbers appear quite limited. At most they can address errors in classes 1 and 2, which account for less than 10% of the errors, and in practice they are likely to address only errors in class 2, which are about 5% of the errors, since in discriminative sequence

models, lexical features are very strong and it is difficult for context to override them.[9] The progress that has been made in the last decade in POS tagging has presumably come mainly from handling some of the cases in class 3, and there is presumably still a fraction of space for improvement here. But many of the cases in class 3 shade off into cases of class 4, where it is hard to imagine a sequence model POS tagger getting them right, except sometimes by a lucky guess. At any rate, classes 3 and 4 together comprise less than one third of the errors. The cases in class 5 are inherently difficult; we return to them at the end. The easiest path for continuing to improve POS tagging seems to be to look at the cases in classes 6 and 7, where the gold standard data is just wrong or is inconsistent because of the lack of clear tagging guidelines. These classes comprise over 40% of the data, and, indeed, if some of the cases that I regard as unspecified or unclear (class 5) could be made clear by tightening up the guidelines, then we might be dealing here with over half the remaining errors. The road on this side of the fence is much less traveled, but I believe it now provides the easiest opportunities for tagging performance gains.

4 Correcting the Treebank

From the earliest days of the resurgence of statistical NLP, there has been a very strong current against fixing data. I think the attitude originated at IBM. For example, one can find a discussion of the issue in David Magerman's thesis [15, p. 101]. I think the idea is that the world is noisy, and you should just take the data as is, in contrast with old-style NLP, which dealt with constructed and massaged data. In addition there are also clear concerns about the overfitting of models, and of model builders being influenced to assign the labels that their models predict. At any rate, one of the big advantages of this perspective is that everyone is using exactly the same training and test sets, and so results are exactly comparable and should be reproducible (after pinning down a few more things about evaluation metrics, etc.).

While it is of course important for everyone to be aware of changes that particular experimenters have made to data sets, and there is certainly value in constant training and test sets for the sake of comparable experiments, it seems that a desire for constancy can be and has been carried much too far. We are now 15 years from the distribution of Penn Treebank release 2, which was the final version of the *WSJ* data, and many researchers have variously noticed mistakes and deficiencies in the annotation, but virtually no attempt has been made to correct them.[10] For example, Ratnaparkhi [16] notes that a large source

[9] But, again, this is for same-epoch-and-domain testing; their impact is much greater when tagging data from disparate domains.

[10] This is not fully true: Work on the PropBank did lead to revisions to and corrections of the Treebank as part of a PropBank-Treebank merge activity, and some other ideas for improving treebank structure (for noun phrase structure and hyphenation) have been incorporated into OntoNotes (LDC2009T24), a new, unified corpus which includes large sections (but not all) of the classic *WSJ* Treebank. However, there

of errors in his tagger is that the tags for certain common words like *about* are inconsistent across the corpus, and, indeed, that the name of the annotator of an example is one of the best predictors of tag assignment. Similarly, Abney et al. [17] examine the most anomalous word tokens that get the highest weights when applying boosting to POS tagging and show that many of these tokens have erroneous tags.

At some point the desire for corpus constancy becomes dysfunctional. The de facto situation with the *WSJ* treebank contrasts with what you see in other fields such as taxonomic biology. It is just not the case that because the first person who collected a certain specimen said it was an Acacia species that for all time it continues to be called an Acacia species, even when further evidence and testing makes it perfectly clear that it is not. At both the individual and species level, the taxonomic biology world has been willing to tolerate quite large scale renamings and disruptions so as to improve the ontological basis of the field. Such is scientific progress in a taxonomic field. The same thing should happen with the content of treebanks.[11]

In computational linguistics, the main work that has been done on improving the taxonomy of tags to allow clearer automatic tagging and improving the conventions by which tags are assigned has been done within the English Constraint Grammar tradition [18, 19]. Contrary to the results above, this work has achieved quite outstanding interannotator agreement (up to 99.3% *prior* to adjudication), in part by the exhaustiveness of the conventions for tagging but also in part by simplifying decisions for tagging (e.g., all *-ing* participles that premodify a noun are tagged as adjectives, regardless). It is surprising the extent to which this work has been ignored by the mainstream of computational linguistics. In some ways, the present work tries to apply some of the same approach to generating consistent taggings, but without performing revisions to the tag set used.

While one way to achieve the goal of correcting the treebank would for humans to carefully check tag assignments, further linguistic annotation work and the developments in language technology provide other methods. A very good way to find errors and inconsistencies in tag assignments is to see where tools like taggers go wrong, as in the examples cited above [16, 17]. Inconsistencies can also be detected by methods aimed just at this task, an idea notably explored by Dickinson [20]. But for the Penn Treebank there is also another profitable approach to pursue. An examination of the corpus makes clear that the *treebanking* was done much more carefully and consistently than the POS tagging. Since, following the ideas of X′ theory, the POS tag of words can often be predicted from phrasal categories, we can often use the tree structure and phrasal

hasn't been correction of a lot of the miscellaneous small-scale errors, and transitioning the community to OntoNotes is still very much a work in progress, with the vast majority of current work on English treebanks and POS tagging and parsing still using the original Penn Treebank Wall Street Journal data.

[11] One venue where this may increasingly happen in the future is in the more open data Manually Annotated Sub-Corpus (MASC) of the American National Corpus: *http://www.anc.org/MASC/*

categories to tell us what the POS tags should have been. This is the main strategy used here to reduce the error and inconsistency rate in the Penn Treebank. While Dickinson's methods are interesting, they do not provide a sufficient level of precision for fully automatic treebank correction, whereas using the treebank syntactic structure commonly does. So, I use testing on the training data to identify inconsistencies, and then information from the treebank structure to guide correction.

5 Fixing some of the errors

Many of the errors and inconsistencies in the Penn Treebank are quite systematic and are well-suited to fixing by deterministic rules. Here, we use Tsurgeon scripts [21], which work by matching a tree pattern using Tregex (a tgrep-like language; cf. [22]) and then performing operations on matched parts of the tree. Here are a couple of cases of the kinds of errors we can straightforwardly fix.[12]

5.1 Past tense versus past participles

There are quite frequent tagging errors as to when verb forms are marked as past tense (VBD) versus past participles (VBN). In general, if a past participle is not adjacent to a passive or perfective auxiliary indicating a VBN, then it is quite frequently wrongly tagged as VBD.[13] But such cases can usually be detected and fixed by rules over tree patterns. For instance, we can use rules such as this one:

> @VP < VBD=bad [> (@VP < (/^VB/ < *be_have_get*)) | > (@VP < CONJP|CC > (@VP < (/^VB/ < *be_have_get*))) | > (@NP < @NP)] relabel bad VBN

That is, a verb in a VP that is under a VP containing a passive or perfective auxiliary verb (perhaps inside a conjunction structure) or modifying a noun phrase should really be a participle VBN and not a past finite VBD.

5.2 Plurals as singulars

The (reasonable) convention for practical part of speech tagging is that words receive a single word class. This flies in the face of commonly accepted ideas of

[12] I will not dwell on the problems caused by hyphenation in the original Penn Treebank, since they are widely recognized and have been reformed in more recent LDC treebanking projects, including the OntoNotes corpus, which contains many of the trees of the Penn Treebank in an updated form which improves the representation of hyphenated terms and complex NPs with left-branching structure.

[13] One could suspect that these errors in many cases reflect a bias resulting from the use of an automatic tagger in the construction of the Penn Treebank, since these were probably cases that the tagger got wrong, and the human annotator then failed to correct.

linguistic morphology where there can be zero derivation of an X^0 category from another X^0 category, with a change in category. There are many such cases in the Penn Treebank, some clearer and some less clear. One case is with plural nouns that become incorporated into named entities which are then treated as singular. Examples include *(the) United States*, *(the) Parks Council*, and *Kawasaki Heavy Industries*. The noun of interest is clearly morphologically plural. But despite the fact that it would normally be regarded as the head of the noun phrase in the first and third examples, the whole noun phrase takes singular verb agreement (*the United states is changing . . .*). Given the stated preference for the Penn Treebank to tag on the basis of syntactic function, it seems like the verbal agreement is a good reason to tag such words as singular NNP, but in practice they are usually – though not consistently – tagged as NNPS. The right answer isn't entirely clear here, involving both how fossilized the formation is and whether to choose the original or final category in cases of X^0 zero derivation. However, in this case, most of the compounds are fairly transparent, and annotators prefer to go with the morphology around three-quarters of the time. Since it is in fashion at the moment to go with the wisdom of crowds, I will adopt this convention.

Another similar problem is when non-nouns become involved in proper nouns, such as *United* in either *United Airlines* or *(the) United States*. It is unclear whether to stick with the adjectival tag for *United* or to call it a proper noun because the whole phrase is clearly a proper noun. In this case, human annotators overwhelmingly went with the NNP choice, and, again, I will follow the wisdom of the crowd. I will summarize these two decisions as the United States Principles.[14] In practice, we can handle cases of these principles simply with rules like:

NNP=bad < Industries|Airlines
relabel bad NNPS

[14] There are further issues here. On proper nouns getting a noun tagging, the Penn Treebank part-of speech tagging guidelines [23, sec 5.3] make a stronger claim, saying that any capitalized word that is part of a name should be tagged NNP or NNPS. This seems too strong, since sometimes verbs and function words are capitalized as parts of names, such as in the titles of books like *Gone With The Wind*. I feel that this is just wrong and very confusing to an POS tagger. Such titles are larger-level syntactic units. While the annotators sometimes followed this dictate, the majority of the time they ignored it. Again, we will follow the wisdom of crowds. This rule will only be applied to content words of a base NP that are part of a name.

Secondly, there are also proper adjectives such as *Australian* or *North Korean*. These are also tagged inconsistently as adjectives or proper nouns in the Penn treebank. Arguably, it is a bad defect of the the Penn Treebank tag set that it lacks a proper adjective category that would cover these cases. But, given the current tag set, when these expressions occur as adjectival modifiers (rather than as a noun referring to a person) then it seems clear that they should be tagged as JJ. These are not proper nouns.

5.3 *That*

That is a hard word for taggers, since it can function as all of a determiner, complementizer, relative pronoun, and an adverb (*he isn't that sick*), and has separate tags for each function (DT, IN, WDT, and RB). Some cases of *that* are clearly ones that need syntactic structure to get right and are beyond the reach of a POS tagger. But there are also lots of errors in the tagging of *that* in the training data, and we may as well at least correct those, so that the POS tagger gets the best chance it can to learn the distinctions. Again, we use the parse structure to guide the correction:

```
@NP < (IN|WDT=bad < /^(?:a|that|That)$/)
relabel bad DT

@SBAR < (DT|WDT|NN|NNP|RB=bad < that|because|while|Though)
relabel bad IN

@ADJP < JJ < (IN=bad < that)
relabel bad RB
```

The first rule matches 173 times in the Penn Treebank, while the second rule matches 285 times. These aren't really rare errors we are talking about.

5.4 Miscellaneous errors and inconsistencies

Many of the details of inconsistencies are particular to individual lexical items and quite mundane. To take one example that turns up a bunch of times, for *K mart*, annotators were inconsistent on whether to tag *mart* as a proper noun or not, presumably because it is not capitalized. It seems to me that it should be treated as still a proper noun, and at any rate, this should just be consistent. This rule make it consistent:

```
@NP < (NNP < K $+ (NN=bad < mart))
relabel bad NNP
```

5.5 Tagging results

Overall, I defined a couple of hundred such rules, based on examination of the training data, some of which changed several hundred tags, others of which changed only a single tag. Some were aimed at outright errors and others at inconsistencies. The rules certainly don't exhaust all the errors and inconsistencies found in the training data, but there are enough covering a number of the most common problem that we can get some idea as to whether such taxonomic improvements might noticeably lift tagging accuracy.

The one complication is how to assess this with respect to test sets. As I show below, if you only correct the training data, then no gains are achieved. This is because the test data has all the same errors and inconsistencies as before.

Indeed, the uncorrected tagger may pick up some of any patterning that exists in "inconsistent" tagging, and do better on the test set. Therefore, the strategy adopted here is as follows: Change rules are developed looking at the training data. These rules are then tested by examining their effect on the development data. It is checked that they do not apply wrongly in any situations (if they do, they are refined to make their application more limited and precise (and the process repeated), or just discarded following Hippocratic reasoning of first doing no harm. Then, the final rules are applied to the final test data without examining their effect. That is, the changes are assumed to all be correct for the test data. Of course, there is a small risk here that a rule could misapply, but the sanctity of the final test data is preserved. Moreover, based on the precise nature of the change rules and examination of their effects on development test data, I feel highly confident that at least 98% of the changes will be good corrections of consistentizations of the test data.

In Table 5 we show the effects of this process on the development data. Note that the number of features in the tagger goes down a bit with data correction because there is less entropy in tag assignments. The error reduction between the first and last lines is already quite substantial (by the standards of these things), and would presumably increase further with further extension and refinement of the correction rule set. Finally, Table 6 show the scores of several models on the final test data.

Table 5. Effect of correction on tagging accuracy on the *WSJ* 19–21 development set.

Model	Corrected Train	Corrected Test	# Feats	Sent. Acc.	Token Acc.	Unk. Acc.
5wShapesDS	no	no	737,955	56.79%	97.28%	90.46%
	no	yes		57.95%	97.38%	90.60%
	yes	no	735,679	55.87%	97.21%	90.58%
	yes	yes		62.66%	97.75%	90.75%

Table 6. Accuracy of taggers on the final test set *WSJ* 22–24.

Model	Corrected Data	Sentence Accuracy	Token Accuracy	Unknown Accuracy
naacl 2003	no	55.75%	97.21%	88.50%
Replication	no	56.44%	97.26%	89.31%
5wShapes	no	56.65%	97.29%	89.70%
5wShapesDS	no	56.92%	97.32%	90.79%
5wShapesDS	yes	61.81%	97.67%	90.49%

6 Foundational Issues

Notwithstanding the significant progress that can be made by removing errors and improving the consistency of the treebank, there are interesting foundational linguistic issues as to which decisions are linguistically well-justified, and which turn into arbitrary conventions of treebank annotation. The latter can still give consistency, but cannot really be linguistically justified.[15]

What I want to look at is the *validity* of what is in the Penn Treebank:

> "Measurement requires three things: An *object* to be measured, a well-defined *property* of the object to measure, and a *measuring instrument* that actually does the job" [24, 135].

The objects at hand here are clear: words and sentences of English newswire. My concerns touch the other two issues: Are part-of-speech labels well-defined discrete properties enabling us to assign each word a single symbolic label? Secondly, is the measuring instrument up to the task? Answering questions like this is one clear place where linguists should have something useful to offer to the modern world of Statistical NLP.

The first question is in many ways the more interesting. If the properties of part of speech and syntactic category are not well-defined, then the variables assigned by the coder lack a coherent basis. Is it possible to assign to each word in a context a single symbol that represents the word's syntactic category? Or do we need something like squishy categories [25]? While the use of discrete categories underlies most of modern generative linguistics, fuzziness is readily accepted by a descriptive grammar such as [26], which regularly refers to the "fuzzy borders between word classes".[16] A thorough recent examination of the issues is found in [27]. Given that the behavior of some words has gradually changed from one part of speech to another over time,[17] some gradable notion of category is presumably necessary. On the other hand, one needs to account for the fact that it seems reasonable and feasible to assign such a category as *noun* or *verb* to the vast majority of the words in the lexicon. This could perhaps be connected up with work on categorical perception [30] which attempts to explore how phenomena which are grounded in continuous physical quantities are perceived by human beings as belonging to discrete categories, with only a little fuzz around the edges.

[15] For instance, consistency could be trivially guaranteed by always giving the same tag to each token of a word type.

[16] Where a treebanker was uncertain concerning the proper part-of-speech tag, they could give words disjunctive tags, and the journal paper [7] describes this as part of a policy of not having annotators make arbitrary decisions. However, in practice, this option was little used, with only 0.01% of tokens (147 tokens) receiving an ambiguous tag. In the vast majority of cases of indeterminacy, it is clear that the annotator either did just make an arbitrary decision or else accepted the decision of the automatic tagger that preceded them. Hence the inconsistency noted in the previous section.

[17] For examples and discussion, see [28], [29], and the discussion below.

What is it that treebankers are actually assigning as categories? [29] showed that many of the criteria that people often use for part of speech are actually sensitive to semantic sorts. Are treebankers mainly influenced by semantic function or are they really picking out structural categories? According to generative wisdom, notional (semantic) criteria for part of speech are "extremely unreliable" [31, 57], but given that they are what is taught in school, if anything ("a noun is a person, place or thing"), there is a high probability that treebankers often use these rather than true syntactic distributional categories. Here I present one example of this phenomenon. I discussed a couple of others in [32].

6.1 Transitive adjectives

Maling [29] discusses the three words *near*, *like*, and *worth*, arguing that these words were historically clearly adjectives, but that with the loss of case marking in English, *like* and *worth* shifted syntactic category to become prepositions (the more appropriate category for uninflected words that take an NP complement), while *near* is perhaps the only surviving case of a transitive adjective in English. In various footnotes, two other candidate surviving transitive adjectives are suggested: *opposite* and *due*. Searching the treebank reveals another possible transitive adjective: *outside*.[18] Table 7 shows a summary of the occurrence of these words in the Penn Treebank Wall Street Journal corpus.

Table 7. Parts of speech assigned to putative transitive adjectives in the Penn Treebank

	Total	IN	JJ	NN	NNS	RB	VB(P)
due	371		344	2	1	24	
like	580	461	26				93
near	126	97	24			5	
outside	145	80	52	8		5	
opposite	19	1	12	6			
worth	114	10	65	39			

The case of *worth* is well-studied. It is a recognized problem word and the treebank manuals have specific, if inconsistent, instructions for it. The initial guide to part of speech tagging said [23, p. 31]:

> *worth* is a preposition (IN) when it precedes a measure phrase, as in *worth ten dollars*.

The subsequent Treebanking manual provides an odd mixture of descriptive and prescriptive advice, but seems to reverse this earlier judgment [33, pp. 308–309]:

[18] The only other word tagged as an adjective and followed by an NP complement is one instance of *such*, but this is because of a clear typo in the newswire source: **Akzo has high hopes for some emerging fiber businesses, such carbon fibers and aramid.*

worth:

1. with complement: ADJP
   Note that some instances of this use of *worth* are labeled PP-PRD, as in (b); however the use of ADJP-PRD, as in (a), predominates.
   (a) [S [NP-SBJ [NP the results], [ADJP however general],] [VP are [ADJP-PRD worth [NP the search]]]]
   (b) [S [NP-SBJ [NP the results], [ADJP however general],] [VP are [PP-PRD worth [NP the search]]]]
2. *dollars worth*: NP
   There is considerable variation, but here is a common way of analyzing expressions like *five dollars worth*:
   [VP issue [NP [NP [ADJP [QP some $ 3 million to $ 4 million] *u*] worth] [PP of [NP Rural Roads Authority bonds]]]]

Commented out in the file is: "Sorry, there ain't no 'right' way for these. –R.". This is the essence of the problem. It is generally accepted that *worth* appears in certain contexts as a noun, but, in the remaining cases, is *worth* a preposition, as Maling and Santorini propose, an adjective as the new Treebanking manual proposes (and also, both the Oxford English Dictionary and Huddleston and Pullum [34]), or should we un-ask this question?

6.2 Treebank evidence

There are 114 instances of *worth*, selectively shown in Table 8. 10 examples are tagged as a preposition, 8 in phrases that treebankers later tagged as ADJPs (1–2) and two that were later tagged as PPs (3–4). In one of the former, the complement is incorrectly tagged as an adverbial (5). 65 examples were tagged as JJ, 48 placed in ADJPs (6–7), 13 placed in PPs (8–9) and 4 which occur inside noun phrases and should have been tagged as NN (10–11). 39 examples were tagged as NN: 2 of these were incorrect and should have been given a non-noun tag (12–13). The rest are noun uses including after a quantifier phrase (14–15) and in other noun uses including compounds (16–17). In 4 cases involving quantifiers (all cases involving a following PP), an extra erroneous level of ADJP structure has been added (18).

There are various questions and concerns here. The OED lists *worth* as a noun, and as an adjective (and as an obsolete verb). [26] appears to regard *worth* as both a preposition and an adjective. On p. 1064 they argue that:

> The prepositional status of *worth* ... is confirmed by the fact that it can govern a noun phrase, a nominal *-ing* clause with a genitive subject, and a nominal relative clause (but not a *that*-clause or a *to*-infinitive

but later (p. 1230) it is listed as a canonical example in the section on "Adjective complementation by an *-ing* participle clause" with an additional note on it being unclear whether to regard *worthwhile* as an adjective or as *worth* followed by a noun (which is reflected in inconsistent spelling). At any rate, they seem to beg the question of the existence of transitive adjectives, by declaring anything

Table 8. Selected citations of *worth* in the Penn Treebank WSJ corpus.

1	Northeast says its bid is	(ADJP-PRD (IN worth)	(NP *e*)).
2	Each share point is	(ADJP-PRD (IN worth)	(NP about $60 million) in sales)
3	grain elevators are	(PP-PRD (IN worth)	(S-NOM *e* preserving for aesthetic . . .))
4	should be	(PP-PRD (IN worth)	(NP 30 a share))
5	assets are	(ADJP-PRD (IN worth)	(NP-ADV more to private buyers than . . .)
6	a good number decide it's not	(ADJP-PRD (JJ worth)	(NP it))
7	and decide it's	(ADJP-PRD (JJ worth)	(NP the astronomical price) to add it
8	It was	(PP-PRD (JJ worth)	it), just for the look on . . .
9	the company . . . is	(PP-PRD (JJ worth)	(NP $70 a share)) if broken up
10	are in need of (NP billions of dollars	(JJ worth)	of repair)
11	is one of the (JJS earliest)	(NN high-net) (JJ worth)	(NNS banks) (PP in the U.S.)
12	Not even . . . makes this trip	(ADJP-PRD (NN worth)	(S taking))
13	What is UAL stock	(ADJP-PRD (NN worth)	(NP *e*))
14	an additional $200 to 300 million	(NN worth)	per month
15	could pile up $150	(NN worth)	of quarters on a slanted coin
16	The company's net	(NN worth)	cannot fall below $185 million
17	thus dilute the	(NN worth)	and voting power of ASKO
18	will sell (NP (ADJP (QP $25 million))	(NN worth)	(PP of his clothes))

with NP complements to be a preposition. Huddleston and Pullum [34] reject the criterion of taking an NP complement as being decisive and come out strongly in favor of *worth* as a transitive adjective, unlike similar words like *like*, *unlike* and *due* which they suggest belong to both the adjective and preposition categories.

Contra Maling, there is some evidence that *worth* is still more like an adjective than a preposition, but it seems fairly clear that it has mixed properties that make it partly like adjectives and partly like prepositions, but not like a canonical member of either category. Even Huddleston and Pullum admit that *worth* "differs markedly from central members of the adjective category". That is, it is a case of syntactic gradience resulting from historical changes [27]. In such cases, it is artificial to demand a categorical classification, whatever its convenience for current part-of-speech tagging technology.

One pragmatic solution in such cases might just be to accept that certain high frequency words may have odd properties and we should just give them tags by convention, however imperfect their assignment to a category. There are probably few applications of NLP which will be much affected by the choice of an adjective or preposition tag for *worth*. If anything, applications are mainly likely to gain from the treatment being consistent. But in such cases, we must accept that we are assigning parts of speech by convention for engineering convenience rather than achieving taxonomic truth, and there are still very interesting issues for linguistics to continue to investigate, along the lines of [27].

Acknowledgments

Thanks to all the people who have worked on the Stanford Part-of-Speech Tagger over the years: foremost, Kristina Toutanova, but also Dan Klein, Yoram Singer, William Morgan, Anna Rafferty, Michel Galley, and John Bauer. And thanks to Alexander Gelbukh for the CICLing 2011 conference invitation, which prompted me to write this material up.

References

1. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: NAACL 3. (2003) 252–259
2. Shen, L., Satta, G., Joshi, A.: Guided learning for bidirectional sequence classification. In: ACL 2007. (2007)
3. Spoustová, D.j., Hajič, J., Raab, J., Spousta, M.: Semi-supervised training for the averaged perceptron POS tagger. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). (2009) 763–771
4. Søgaard, A.: Simple semi-supervised training of part-of-speech taggers. In: Proceedings of the ACL 2010 Conference Short Papers. (2010) 205–208
5. Subramanya, A., Petrov, S., Pereira, F.: Efficient graph-based semi-supervised learning of structured tagging models. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. (2010) 167–176
6. Collins, M.: Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In: EMNLP 2002. (2002)
7. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics **19** (1993) 313–330
8. Finkel, J., Dingare, S., Manning, C., Nissim, M., Alex, B., Grover, C.: Exploring the boundaries: Gene and protein identification in biomedical text. BMC Bioinformatics **6 (Suppl 1)** (2005)
9. Collins, M.: Ranking algorithms for named entity extraction: Boosting and the voted perceptron. In: ACL 40. (2002) 489–496
10. Tsuruoka, Y., Tsujii, J.: Bidirectional inference with the easiest-first strategy for tagging sequence data. In: Proceedings of HLT/EMNLP 2005. (2005) 467–474
11. Clark, A.: Combining distributional and morphological information for part of speech induction. In: EACL 2003. (2003) 59–66
12. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL 41. (2003) 423–430
13. MacKinlay, A.: The effects of part-of-speech tagsets on tagger performance. Honours thesis, Department of Computer Science and Software Engineering, University of Melbourne (2005)
14. Church, K.W.: Current practice in part of speech tagging and suggestions for the future. In Mackie, A.W., McAuley, T.K., Simmons, C., eds.: For Henry Kučera: Studies in Slavic philology and computational linguistics. Number 6 in Papers in Slavic philology. Michigan Slavic Studies, Ann Arbor (1992) 13–48
15. Magerman, D.M.: Natural language parsing as statistical pattern recognition. PhD thesis, Stanford University (1994)
16. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: EMNLP 1. (1996) 133–142
17. Abney, S., Schapire, R.E., Singer, Y.: Boosting applied to tagging and PP attachment. In Fung, P., Zhou, J., eds.: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. (1999) 38–45
18. Voutilainen, A., Järvinen, T.: Specifying a shallow grammatical representation for parsing purposes. In: 7th Conference of the European Chapter of the Association for Computational Linguistics. (1995) 210–214
19. Samuelsson, C., Voutilainen, A.: Comparing a linguistic and a stochastic tagger. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. (1997) 246–253

20. Dickinson, M., Meurers, W.D.: Detecting errors in part-of-speech annotation. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03). (2003)
21. Levy, R., Andrew, G.: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In: 5th International Conference on Language Resources and Evaluation (LREC 2006). (2006)
22. Rohde, D.L.T.: Tgrep2 user manual. MS., MIT (2005)
23. Santorini, B.: Part-of-speech tagging guidelines for the Penn treebank project. 3rd Revision, 2nd printing, Feb. 1995. University of Pennsylvania (1990)
24. Moore, D.S.: Statistics: Concepts and Controversies. 3rd edn. W. H. Freeman, New York (1991)
25. Ross, J.R.: A fake NP squish. In Bailey, C.J.N., Shuy, R.W., eds.: New Ways of Analyzing Variation in English. Georgetown University Press, Washington (1973) 96–140
26. Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: A Comprehensive Grammar of the English Language. Longman, London (1985)
27. Aarts, B.: Syntactic gradience: the nature of grammatical indeterminacy. Oxford University Press, Oxford (2007)
28. Abney, S.: Statistical methods and linguistics. In Klavans, J., Resnik, P., eds.: The Balancing Act. MIT Press, Cambridge, MA (1996)
29. Maling, J.: Transitive adjectives: A case of categorial reanalysis. In Heny, F., Richards, B., eds.: Linguistic Categories: Auxiliaries and Related Puzzles. Volume 1. D. Reidel, Dordrecht (1983) 253–289
30. Harnad, S., ed.: Categorical perception : the groundwork of cognition. Cambridge University Press, Cambridge (1987)
31. Radford, A.: Transformational Grammar. Cambridge University Press, Cambridge (1988)
32. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Boston, MA (1999)
33. Bies, A., Ferguson, M., Katz, K., MacIntyre, R., and colleagues: Bracketing guidelines for Treebank II style: Penn treebank project. MS, University of Pennsylvania (1995)
34. Huddleston, R.D., Pullum, G.K.: The Cambridge Grammar of the English Language. Cambridge University Press, Cambridge (2002)