POS Tagging with a More Informative Tagset

Andrew MacKinlay[†] and **Timothy Baldwin**[‡]

[†‡] Dept. of Computer Science and Software Engineering
University of Melbourne
Victoria 3010 Australia

[‡] NICTA Victoria Research Lab
University of Melbourne
Victoria 3010 Australia

{amack,tim}@cs.mu.oz.au

Abstract

We investigate the impact of introducing finer distinctions into the tagset on the accuracy of part-of-speech tagging. This is a tangential approach to most recent research in the field, which has focussed on applying different algorithms using a very similar set of features. We outline the basic approach to tagset refinement and describe preliminary findings.

1 Introduction

Most recent research on corpus-based part-of-speech (POS) tagging has tended to focus on applying new algorithms to an existing task (Brill, 1995; Ratnaparkhi, 1996; Giménez and Màrquez, 2004), or improving the efficiency of an existing algorithm (Ngai and Florian, 2001). While there has been some successful experimentation with modifying the feature sets of particular taggers (Toutanova and Manning, 2000), the various state-of-the-art taggers for the most part use a very similar set of features in determining the tag for a particular token: some subset of the two preceding and two following tokens, and their tags. The different algorithms have tended to plateau to a similar "glass ceiling" in accuracy ($96.9 \pm 0.3\%$ over all tokens for the taggers in this paper) by using these features.

POS tagging is essentially an optimisation process over firstly the tag sequence and secondly the tag–word assignments for a given input. The relative difficulty of this task hinges on the internals of the POS tagset, and the tagging performance over a given dataset can vary greatly depending on the tagset used. In this paper, we seek to enhance tagging performance by adding a third dimension to the optimisation process: the tagset. We explore the possibility of transforming the tagset via reversible (lossless) mappings, to produce a dataset which is more amenable to automatic tagging and thus results in higher performance than the original tagset. We follow the majority of recent mainstream research on English POS tagging in adopting as a baseline the tagset used in the Penn Treebank (Marcus et al., 1993).

We evaluate two different approaches to identifying patterns of syntactic regularity with the existing POSs. Our primary approach is linguistic insight: we investigate a range of linguistically motivated subdivisions which are either designed to assist in a specific problematic instance of disambiguation, or are linguistically sensible enough to be applied for independent reasons. Additionally we compare this with a data-driven approach, where we attempt to identify intra-POS groupings by running a clustering algorithm over the words within a particular class using features derived from syntactic context. We report the most promising results achieved in both cases.

Section 2 outlines some diverse algorithms which have been applied to POS tagging; Section 3 gives some motivation for attempting increases in accuracy; Section 4 describes details of the tagset used in the Penn Treebank; Section 5 outlines our method; in Section 6 we show results for various strategies and in Section 7 we discuss further work.

2 Tagging Algorithms

A large number of algorithms have been applied to POS tagging; a brief treatment of those which are relevant follows.

2.1 Transformation-Based POS tagging

The transformation-based learning (TBL) paradigm as applied to POS tagging was first described in Brill (1995); like all of the taggers described here it is a corpus-based method. In the learning phase, a TBL tagger assigns each word the most-likely unigram tag from the training data, and generates a large set of possible transformational rules which

Proceedings of the Australasian Language Technology Workshop 2005, pages 40–48,
Sydney, Australia, December 2005.

map the unigram tagger assignments onto the gold-standard assignments, conditioned on contextual word and tag features. It iteratively selects from these the rule which minimizes the number of errors, and applies that rule to modify the assigned tags. The output is an ordered list of rules which can then be applied, in combination with the learned unigram tag probabilities, to unseen data.

The TBL implementation used here is fnTBL 1.1 (Ngai and Florian, 2001); it is equivalent in power to Brill's original but runs two orders of magnitude faster due to optimisations which are not relevant here. The reported accuracy in Brill (1995) was 96.6%/81.2% for known/unknown words using 1M words of the Penn Treebank WSJ Corpus as training data and 200K words as test data.

2.2 Maximum Entropy POS tagging

The maximum entropy framework is a probabilistic approach to NLP commonly used for classification tasks including POS tagging. The approach was applied specifically to POS tagging in Ratnaparkhi (1996). The underlying principle is that when choosing between a number of different probabilistic models for a set of data, the most valid model is the one which makes fewest arbitrary assumptions about the nature of the data.

The probabilistic information in this case comes from a set of binary-valued features which in Ratnaparkhi (1996) are dependent solely on local contextual features: the current word and the two words on either side, and the two preceding tags. In Toutanova and Manning (2000) a number of other hand-tuned features derived from a larger context window are added to assist in disambiguation in problematic words, and activated only upon the occurrence of such words. These optimisations bring the accuracy from the baseline for all/unknown words of 96.76%/84.5% (using a subset of the feaures in Ratnaparkhi (1996)) to 96.86%/86.91%.

2.3 SVM POS tagging

Support vector machines (SVMs) have been applied to POS tagging in Giménez and Màrquez (2004), inter alia. The features are parallel to those used in a maximum entropy model: a set of binary features conditioned on the presence of words and tags within a local context window. These features are then used to build an SVM for each part of speech which contains ambiguous lexical items (reportedly 34 for the Penn Treebank WSJ corpus), and in the classification stage, the most confident prediction from all of the SVMs is selected as the tag for the word. The accuracy reported is 97.16%/89.01% for all/unknown words.

3 Motivation

As noted in Garside et al. (1997), the linguistic quality of a tagset is determined by the extent to which each tag denotes a set of words with a unique set of common syntactic properties, while the computational tractability of it is determined by the ease with which the tag for a particular token can be determined, and how much each tag aids in the disambiguation of surrounding words. The extreme cases of tagsets with either one tag per word or one tag for all words, are examples of tagsets which are highly tractable in computational terms, but of very little use linguistically, which perhaps serves to indicate that these requirements sometimes conflict. However, the aim here is to test whether there is always an inverse relationship between the two. A tagset which encodes more subtle distinctions is almost inevitably more useful in linguistic terms unless the additional distinctions are entirely random; here we will test whether the accuracy can be increased by certain carefully selected tagset subdivisions motivated by linguistic intuition.

Indeed, in Klein and Manning (2003) it was demonstrated that a finer-grained set of category labels can markedly improve performance in the related application of parsing, by providing more contextual information upon which to base decisions in cases of ambiguity. This, along with the demonstration by Toutanova and Manning (2000) that there is potential to improve POS tagging performance by adding linguistically motivated features to the tagger suggests that it may be possible to apply an analogous version of Klein and Manning's method to POS tagging. If we alter the tagset to encode more subtle distinctions within the word classes, these new divisions could potentially increase the computational tractability of the tagset and hence improve the performance of the tagger, since subtler distinctions can provide more useful information to disambiguate surrounding words.

It is worth addressing the question here of why it is worth striving for a small performance improvement here. By NLP standards, accuracy of ∼97.0% seems astoundingly high, begging the question of whether there is any point in attempting to raise this figure by a few fractions of a percent. However, according to word-by-word evaluation metrics, POS tagging is actually quite a simple task – as noted by Charniak et al. (1993), the unigram-based most-likely tag (MLT) baseline for the task is around 91%.

The problem is POS tagging is generally a pre-processing phase in NLP, which acts as input to a second stage such as sentence-level parsing. If we look at sentence-level accuracy i.e. the proportion of sentences in which all tokens are correctly tagged, the POS tagging task seems harder – with an average sentence length of ∼24 words and assuming errors occur independently we would expect a tagger which gives 97% accuracy over word tokens to achieve 49% at the sentence level, while a tagger performing at 98% should tag 62% of sentences correctly.

4 The Penn Treebank Tagset

The tagset for the Penn Treebank is based on the tagset used for the original Brown corpus (Francis and Kučera, 1979) but at 36 tags (excluding punctuation), it is small in comparison to both the Brown tagset (75 non-compounded tags[1]), and other related tagsets. This was a deliberate design decision, in that Marcus et al. (1993) set out to create "A Simplified POS Tagset for English" to alleviate problems of sparse data in stochastic applications – thus increasing the computational tractability of the tagset. The primary means by which they achieved this simplification was with by applying the notion of 'recoverability': if the distinctions between several tags could be recovered from either syntactic information (available from the parse tree annotations) or lexical information (the character string making up the word), the tags could be conflated.

The avoidance of lexically recoverable distinctions means that classes with just a single lexical item are dispreferred – hence, for example, the abandonment of the explicit POS distinction between auxiliary verbs and content verbs which is made in most other tagsets derived from the Brown tagset (Francis and Kučera, 1979; Garside et al., 1987; Garside et al., 1997). Additionally the presence of syntactic information means that the traditional distinction between prepositions and subordinating conjunctions can be removed as it can be recovered from the phrasal category of the sibling (*SBAR* for a subordinating conjunction and *NP* for a preposition).

However Marcus et al. (1993, p315) stress that all of this information is available to users of the corpus via additional sources:

> ...the lexical and syntactic recoverability inherent in the POS-tagged version of the Penn Treebank corpus allows end users to employ a much richer tagset than the small one described ... if the need arises.

What is interesting here is that the tagset was not designed to differentiate all possible distributional differences when other information is available, but in examples of POS tagging in the literature, the tagset is invariably used in unaltered form despite the tagger having no explicit access to the syntactic information required to recover sub-usages of a given tag.

The lexical information is used, albeit implicitly, by the inclusion of lexicalised features in all of the state-of-the-art taggers mentioned here. Ironically, the Penn Treebank tagset was designed to be coarse to avoid problems of data sparseness, and yet it is this coarseness which contributes to the inevitable inclusion of sparsely-populated lexicalised features to achieve high accuracy. While there have been examples of certain ad hoc modifications, Manning and Schütze (1999) note that a systematic study into the effect of the tagset has not been explored. It seems the possibility of making explicit certain syntactic regularities within the coarse Penn Treebank word classes for the purposes of improving performance in POS tagging is one worth investigating.

5 Method

We wish to investigate here whether we can improve performance by helping the tagger make syntactic generalisations which are not apparent either from the coarse POS tags or from the sparsely populated lexical feature vector. Subdividing the tags in a linguistically sensible way should hopefully provide this information. However the presence of additional POSs clearly has the potential to make the POS tagging task more difficult. Thus, as shown in Figure 1, we will map the tag of each token in the training

[1] For comparison with the Penn Treebank, where the 's suffix is split from the host noun, this figure excludes 12 possesive variants of other tags such as *NN$*

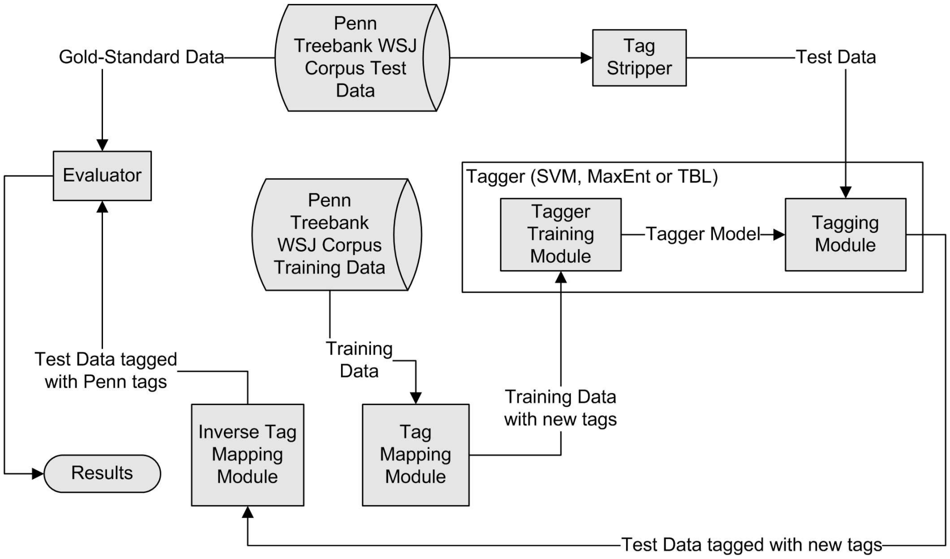
Gold-Standard Data
Penn Treebank WSJ Corpus Test Data
Tag Stripper
Test Data
Evaluator
Tagger (SVM, MaxEnt or TBL)
Penn Treebank WSJ Corpus Training Data
Tagger Training Module
Tagger Model
Tagging Module
Training Data
Test Data tagged with Penn tags
Training Data with new tags
Inverse Tag Mapping Module
Tag Mapping Module
Results
Test Data tagged with new tags

Figure 1: The experimental architecture

data appropriately to a particular new version of the tagset, run the trained tagger over a test corpus, and for purposes of comparison map the finer-grained POS-tags back to the original Penn Treebank tags before evaluating performance. This method means that any increased linguistic utility of the mapped tags is discarded before evaluation, but for the purposes of this experiment the linguistic utility is a means for improving tagger performance rather than an end in itself.

To facilitate the final stage of mapping the tags back to original Penn tags, we place certain restrictions on allowable modifications: the mapping function must be either be injective from the old to the new tags, or any distinctions which are collapsed must be unambiguously recoverable from the wordform so the equivalent tags from the original tagset can be determined reliably.

Our tag-mapping module enables any subset of POS tag assignments to be translated using a conjunctive or disjunctive combination of lexical and syntactic features. Syntactic features include the two surrounding tags, and the phrasal categories of nearby nodes (as defined within the treebank annotation): parent, grandparent, immediately preceding or following siblings, or all preceding or following siblings.

Initial experiments suggested that the marginality of the performance improvements we were aiming for over the data meant that there was a risk of overfitting – even with 200K words of test data, a global change of 0.05% corresponds to only 100 words, or much less over a specific POS; additionally, the inter-annotator discrepancies noted in Ratnaparkhi (1996) are likely to swamp any corpus-wide generalisations. To alleviate this, we used five-fold cross-validation over sections 00-22 of the Penn Treebank WSJ corpus, effectively producing a development set of ∼1M words. Rather than split the data by sections, the data partitions were constructed by placing one sentence from every five in each partition. This tends to inflate performance figures, however this is not a problem here since we are purely looking for improvements relative to the benchmark.

We selected fnTBL (Ngai and Florian, 2001) as our first stage prototyping tool for a set of tagset modifications, as it can complete a five-fold cross-validation test-cycle in under two hours. Any modification which had a large negative impact on performance at this stage was generally not investigated further, since the taggers use a similar set of features, and we were attempting to find universally useful distinctions. The SVM tagger SVMTool 1.2.2 (Giménez and Màrquez, 2004), with a turnaround of under seven hours, was used in subsequent experimentation. Only the Stanford NLP Maximum Entropy tagger (Toutanova and Manning, 2000) had a prohibitive training time, so for practical reasons was used minimally, for benchmarking and later-stage testing.

5.1 Evaluation Metrics

We evaluate the results using several evaluation metrics. First, for comparison with previous work we use the global token-level accuracy metric since it is the most widely-used metric in tagging research. The token-level accuracy over unknown words (i.e. those which did not appear in the training data) is also crucial since this is a major source of tagging errors – in our baseline with an unmodified tagset, just 2.4% of the tokens in the training data were unknown but they contributed 11-13% of errors. Additionally, we show sentence-level accuracy, and precision, recall and F-score over individual POSs.

5.2 Sources of modified tagsets

The primary goal here is to apply linguistic intuition to the task of tagset modification. Potential modifications were drawn from a number of sources, including grammars of English (such as Huddleston and Pullum (2002)) and alternative tagsets, such as CLAWS7 (Garside et al., 1997), and evaluated empirically.

An alternative line of investigation was more data-driven: we investigated whether in a separate stage to training the taggers, we could use machine learning techniques to determine useful subdivisions in the tagsets. To this end, we defined a range of features which could help in determining patterns of syntactic regularity. Some of the features were syntactic, often corresponding to layers of annotation used in Klein and Manning (2003): phrasal categories of the parent, grandparent, left sibling and right sibling, and binary-valued features for whether a given preterminal corresponds to a phrasal head, or whether it is the only element in its phrase. There were also a set of collocational features corresponding more closely to the features available to the tagger, based on the two preceding and two following POSs.

The nominal values of each feature were extracted for each token in the training/development data then conflated by word type and converted into a frequency distribution across the possible feature values for each word type. The value distribution for each feature with n non-zero values was then converted into a set of n real-valued features for the word type using maximum likelihood estimation. This method of combining feature values is not ideal but was the most principled way we could find of capturing a large amount of distributional information manageably.

These feature values were then used as input for the implementation of the EM algorithm in the Weka toolkit (Witten and Frank, 2000). Several different combinations of features were used; broadly, they were syntactic-only, collocational-only and both.

6 Results

6.1 Baseline and Benchmark

The benchmark results from running each of the publicly available taggers with the default or recommended parameter settings are shown in Table 1, with results over specific POSs in Table 2. For a point of comparison, we also applied a suite of naively conceived modifications to illustrate the effects of data sparseness. The idea is borrowed from POS induction, which involves determining word clusters (i.e. POSs) from unannotated data. The task here is similar except that we are looking for patterns of regularity within a particular POS, so the baseline used in Clark (2003) may be informative. To subdivide a part of speech into n subclasses, we assign each of the $(n-1)$ most frequently seen word tokens from the class into $(n-1)$ separate new classes and the remainder to a final subclass. In Table 3, we present the results for $n = 2, 3, 4$ over closed-class POSs of reasonable size, after training and tagging with fnTBL using the same broad indicators shown in Table 1. The best-performing modification from this selection, (i.e. for subdividing *PRP*, with $n = 1$) was additionally tested using SVMTool and the Stanford MaxEnt Tagger; these results are shown in Table 1.

6.2 Linguistically Motivated Modifications

We present here the results for a selection of linguistically motivated modifications which were most successful or most motivated from a theoretical point of view. One obvious candidate modification is reversing the idiosyncratic conflation of prepositions and subordinating conjunctions in the Penn Treebank. This could have been achieved lexically, by extracting a list of lexemes which frequently act as subordinators in the training data, and mapping the tags of the tokens accordingly. However, the most successful and principled approach was using syntactic features for each token and thus deciding on a word-by-word basis. This captures the fact that there are certain words such as *before* that are ambiguous between the two; we let

	TBL			SVM			MaxEnt		
	All	Unk	Sent	All	Unk	Sent	All	Unk	Sent
Benchmark	96.842	81.94	51.77	96.852	84.62	50.72	97.056	87.34	53.72
Freq-based *PRP*:2	96.839	81.81	51.69	96.851	84.60	50.67	97.048	**87.40**	53.51
Freq-based *RB*:3	96.843	81.73	51.72	96.855	84.67	50.71	97.056	87.28	53.72
Clust: IN, All	96.831	81.79	51.52	**96.865**	84.64	**50.90**	**97.065**	87.32	**53.78**
Clust: IN, Coll	**96.850**	**82.00**	**51.82**	96.855	84.61	50.74	97.050	87.32	53.59
Ling: IN—SUB	96.842	81.76	51.63	96.855	84.65	50.77	97.050	87.37	53.51
Ling: RB–DEG	96.818	81.66	51.69	96.847	**84.72**	50.63	–	–	–
Ling: IN—RP	96.832	81.59	51.51	96.851	84.63	50.73	–	–	–

Table 1: Accuracy (%) of the best-performing or most motivated tag modifications for each of the broad methods discussed using five-fold cross-validation over sections 0–22 of the WSJ corpus, with the highest accuracy figure in each column in bold

	TBL		SVM	
	All	Unk	All	Unk
JJ	91.66	76.01	92.22	80.84
JJR	87.55	34.86	88.41	41.44
JJS	93.26	73.87	95.46	70.33
NNPS	65.6	20.78	62.62	19.65
RBR	70.47	–	71.86	–
RBS	78.55	–	86.04	–
VBD	95.06	72.25	95.46	75.25
VBP	92.97	55.46	93.06	44.44

Table 2: Benchmark F-Score (%) over 1,047K words of text, for selected POSs

the tagger resolve this ambiguity as appropriate. Two syntactic features were used to determine if a given *IN* token is a subordinating conjunction (preposition being the default): an *SBAR* parent node or an *S* immediate right sibling.

The results for SVMtool showed very few differences for recall and precision over individual POSs compared to the benchmark: the largest change was a 2% relative increase in F-score over unknown *VBP*s (verb, present tense, non third person singular) and a 3% relative increase for unknown *JJR*s (comparative adjective). The results for fnTBL in comparison were more varied, with a 7% increase in F-score over known members of *RBS* (superlative adverb), and for unknown words, an 8% decrease for *VBP*s and a 5% increase for *JJR*s. The changes in *JJR* are probably due to *than*, which often occurs in their vicinity (e.g. *higher than*) and is usually a preposition by our definition but tends to occur in different contexts to subordinating conjunctions such as *because*. The other differences are harder to account for, and are perhaps unpredictable outcomes due to data sparseness.

Another candidate modification is based on the observation that in the baseline taggers, 5.8-6.4% of tagging errors were due to a gold-standard *JJ* (adjective) being tagged *VBN* (verb past participle) or vice versa, with a further 1.9-2.0% of errors due to the corresponding *JJ*/*VBG* (verb present participle) confusion. This distinction is notoriously difficult to make, but we should be able to assist in discrimination by utilising the linguistic tests distinguishing between the two: adjectives can be modified by degree adverbs such as *very*, while verbs cannot. Thus, the presence of a degree adverb should indicate unequivocally that the head word is an adjective. In reality there is no clear boundary between degree adverbs and the more common verb-modifying adverbs, and empirically the most effective approach, as with the IN–SUB modification, was to allow ambiguity of degree adverb membership and condition the tag mapping on syntactic features for each token: an *RB* with either an *RB* or *JJ* as its right sibling, or an *ADJP* (adjective phrase) as parent was mapped as a degree adverb. This modification is denoted RB–DEG in Table 1.

Compared to the benchmark, the results for SVMtool were again reasonably similar to the baseline, with the only significant differences in F-score being over unknown words: increases of 2% for *JJS* and *VBP*, and 3% for *VBD* which were offset by decreases of 7% for *JJR*. fnTBL over known words gave a 33% relative decrease in F-score for members of *RBS*, and a 5% decrease for *JJS* (superlative adjective), while over unknown words the largest changes in F-score were a 54% increase for *NNPS*, a 3% increase for *VBP*, as well as a 6% decrease for *JJR*. The

POS	IN			DT			PRP		
n	2	3	4	2	3	4	2	3	4
All Tokens	96.823	96.817	96.819	96.806	96.813	96.806	96.839	96.830	96.838
Unknown	81.40	81.78	81.57	81.61	81.78	81.64	81.81	81.95	81.71
Sentences	51.41	51.28	51.55	51.13	51.38	51.28	51.69	51.57	51.60

Table 3: Overall accuracy (%) with naively subdivided POSs using fnTBL

	TBL			SVM			MaxEnt		
	All	Unk	Sent	All	Unk	Sent	All	Unk	Sent
Benchmark	96.68	83.71	49.52	96.75	87.23	49.76	96.99	88.50	52.92
Clust IN, All	96.68	83.59	49.91	**96.78**	**87.38**	**50.04**	96.990	88.47	52.88
IN–SUB	96.70	84.07	**50.00**	96.77	87.32	49.94	96.971	88.29	52.70
VB–INF	**96.73**	**84.10**	49.94	96.75	87.26	49.74	–	–	–

Table 4: Accuracy (%) of selected tag modifications from Table 1 over the held-out 129K-token test set of sections 22 and 23 of the WSJ corpus with sections 0–22 as training data

RBS/*JJS* differences are probably due to confusions between each other for *most* which is often ambiguous when preceding prenominal adjective phrases (e.g. *the most ethical policies*), and *RB-DEG*s which occur in such *ADJP*s lead to spurious generalisations. The differences over unknown *JJR* are probably due to 'degree adverbs' (by our syntactic criteria) such as *much* which modify comparative *ADJP*s and operate quite differently to words such as *very*. Again, we must assume some changes are due to unpredictable data sparseness.

A further round of tests was designed to increase computational tractability with little reference to linguistic motivation. It concerns the ambiguity between *IN* and *RP* (particle). Again, these POSs are notoriously difficult to distinguish between, since many words such as *on* are systematically ambiguous between the two. However, there are many members of *IN* which have no homographs in the *RP* class. If we map the ambiguous members of *IN* to a new class, we are explicitly indicating to the tagger whether or not a word is ambiguous between the two POSs and could improve performance for these particular words. Interestingly, this modification, denoted IN–RP, achieved better performance when applied in conjunction with the IN–SUB modification mentioned (96.832% accuracy over all tokens) above than when it was used alone (96.818% over all tokens).

Various other modifications included retagging verbs based on their likely complements (e.g. if its complement is likely to include a particle, bare infinitive or noun phrase), and several sets of modifications for adverbs, including locative adverbs and those homographic with prepositions. Resultant accuracy using fnTBL ranged from roughly equal to the reported figures to 0.3% below them.

6.3 Intra-POS Clustering Modifications

After running the clustering algorithms with different feature sets as input, we selected a large range of promising sets of POS clusters with a qualitative examination of the clustering output, then starting with fnTBL we successively narrowed down the set of clusters tested with each algorithm by selecting the more successful modifications for the next stage (SVMTool), until finally testing the best-performing modification with the Stanford Maximum Entropy Tagger.

The best performing modification came from using all of the syntactic and collocational features mentioned and resulted in splitting *IN* into four subclasses, corresponding roughly to transitive prepositions (this however included some types such as *before* which can be used as subordinators), rare prepositions, subordinators and a cluster containing only *than*. We also show results for another effective clustering, which again dealt with *IN* using only collocational features. The clusters here do not show such a discernible pattern. In both cases, but particularly the latter, we suspect overfitting due to the fact that the statistics for clustering were derived from the entire combined training/development set.

6.4 Final Testing

To evaluate the validity of our suspicions of overfitting by the clustering algorithm we also show in Table 4 a final round of testing using sections 0-21 as training data and sections 22-23 (which had been held out until this point and were not used to generate clusters) as a test set. This also facilated comparison between the linguistically motivated modifications and the clustering modifications. We would expect the linguistically motivated modifications, which were generated in a fairly data-independent manner (apart from the selection of different modifications on the basis of performance over the development set) to display more consistent improvments over held-out data than the data-driven clusters.

7 Discussion

It is clear from the results shown here[2] that to an extent the intuitions of Marcus et al. (1993) about data sparseness were justified. Table 3 shows that coming up with a modification which reduces performance is easy; we have demonstrated here that finding a set of non-detrimental modifications is difficult. There are probably several reasons for this. It is the most difficult 3% of tokens which we are attempting to tag correctly. Among these are words which probably cannot be tagged correctly with a small context window, words for which humans would have difficulty agreeing on a tag, and words which are tagged incorrectly in the gold standard (a fact which was explored in Ratnaparkhi (1996)).

However despite this, there are still reasons to believe that there is room for improvement. As noted in Brill and Wu (1998), there is high degree of complementarity in errors made by maximum entropy and TBL-based taggers (among others), suggesting that even though these taggers use similar contextual features, the differences in the way these features are combined result in errors over different words. This tends to imply that at least some of the time, there is sufficient information available, but that the different underlying algorithms fail to apply it correctly in all cases.

Given this, the lack of success so far in applying linguistic intuition was surprising. While the highest-performing modification was the linguistically-motivated reintroduction of subordinators, accuracy in this best case was not significantly different from using an unmodified tagset. However the worst of the linguistically motivated modifications resulted in markedly lower accuracy than the benchmark. Even modifications targeted at addressing a specific confusion (such as RB–DEG) actually reduced performance. Additionally, most of these linguistic modifications were outperformed by the best naive frequency-based approach.

The clustering was not designed on a particularly firm theoretical basis; rather, we attempted it as a comparison with the linguistically motivated methods. Despite this, it has produced some intra-POS clusters which (slightly) improve performance, however some of this may be due to overfitting. The performance over the test set, at least for SVMTool, could be seen to support the validity of the result. However examining the output from all three taggers together shows there is very little evidence of consistent improvement from any individual mapping. While they can produce slight improvements for certain taggers in certain cases, these improvements are not significant, and there is little firm evidence on the basis of this experiment for the significant utility of either the data-driven or linguistic approaches.

It is apparent from Table 1 that the best results from various methods seem to asymptote towards the benchmark using the unmodified tagset, which is indicative of the inherent difficulty of the task. Even when we make justifiable modifications, the increased data sparseness usually results in a net performance decrease. While we would not rule out improved results from this line of experimentation, it is likely at least that some variation on the strategy will be necessary for an appreciable increment in tagging accuracy.

Possible further strategies we plan to investigate include adding a two-tiered classification system, by systematically adding delimiters to newly created tags, and adding contextual features dependent on the portion of the POS tag preceding or following the delimiter. Multiple levels of classification of POS tags are used successfully in the JAWS tagging system (Garside et al., 1997) but do not appear to have been applied to the the Penn Treebank. This method would give the taggers access to the more densely populated coarse-tag features when necessary, but when the subtler distinc-

[2] For a more extensive set of results which support the same conclusion, as well as a more detailed discussion of methodology, see MacKinlay (2005)

tions we have added are useful the taggers can utilise them. This is of course a question requiring further experimentation.

References

Eric Brill and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 1, pages 191–195, Montreal.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–65.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. Equations for part-of-speech tagging. In *National Conference on Artificial Intelligence*, pages 784–789.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of EACL'03: 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 00–00, Budapest, Hungary.

W. N. Francis and H. Kučera. 1979. Brown Corpus manual: Manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. Brown University, Providence, Rhode Island, USA.

Roger Garside, Geoffrey Leech, and Geoffrey Sampson, editors. 1987. *A Computational Analysis of English.* Longman Group UK, Essex, England.

Roger Garside, Geoffrey Leech, and Anthony McEnery, editors. 1997. *Corpus Annotation: Linguistic Information from Computer Text Corpora.* Addison Wesley Longman Ltd, New York, USA.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Rodney Huddleston and Geoffrey K. Pullum, editors. 2002. *The Cambridge Grammar of the English Language.* Cambridge University Press, Cambridge, UK.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

Andrew MacKinlay. 2005. The effects of part-of-speech tagsets on tagger performance. Honours thesis, University of Melbourne.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, MA.

Mitchell P Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Grace Ngai and Radu Florian. 2001. Transformation-based learning in the fast lane. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 40–7, Pittsburgh, USA.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Somerset, New Jersey. Association for Computational Linguistics.

Kristina Toutanova and Christoper D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, Hong Kong, China.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, San Francisco, CA.