Training Set Properties and Decision-Tree Taggers: A Closer Look

Brian R. Hirshman
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
hirshman@cs.cmu.edu

Abstract

This paper examines three ways to improve part-of-speech tagging accuracy: by increasing the number of training examples presented to the tree learner, by increasing the number of word-specific subtrees grown, and by increasing the number of ngrams (preceding parts of speech) per training example. Though experimental results indicate that additional training data generally leads to the greatest amount improved accuracy, they also demonstrate that including word-specific subtrees can be useful and that trees considering two or more previous parts of speech in their classification decision are superior to those examining just one.

1 Introduction & Motivation

Part-of-speech tagging, the process of assigning a grammatical tag to a given word, codifies how a word is used in a sentence [1]. These codes provide clues that help readers and listeners understand the meaning of the word [1]. Past research in part-of-speech tagging has been relatively successful at predicting tags in test data [1, 2]. Many different types of learners have demonstrated that it is possible to use large amounts of training data in order to achieve accuracies well above ninety percent [1, 2, 3, 4].

One common technique for predicting part-of-speech tags is decision tree learning [4, 5]. A decision tree learner is a machine learning algorithm that sequentially partitions the training data set based on attribute values, recursively partitioning the data until all attributes at a node can be classified with the same value [6]. Unigram, bigram, trigram, or quadrogram trees (trees using the one, two, three, or four immediately previous words) have been able to predict upcoming parts of speech with high accuracy on very large training sets [4]. When augmented with special rules to tag parts of speech based on word suffixes, these taggers have achieved accuracy rates of nearly 96% [4].

This work modifies the data training set in three ways and examines the effects of these modifications on the accuracy of part-of-speech prediction. First, this paper examines the most straightforward change to the data set, providing additional labeled training examples. This additional training data can be used to grow a better tree. Additionally, this paper seeks to evaluate the improvement that comes from increasing the number of word-specific subtrees. By creating special subtrees for common words, decision tree learners may be able to boost accuracy rates. Lastly, this paper examines the effect of increasing the number of previous parts-of-speech examined when making a classification decision. By examining additional past words, a decision tree may have more contextual information on which to base its decision.

2 Problem Definition

This paper seeks to address three issues with respect to decision-tree part-of-speech taggers:

- What effect – if any – occurs from increasing the number of examples in the training set? At what point is impractical for reasons related to lack of improvement or computational infeasibility?
- What effect – if any – occurs when subtrees are grown to account for specific words? How do modifications to the number of these subtrees affect the maximum accuracy rate, and how does it affect the rate of change in the accuracy rate?
- What effect – if any – occurs from increasing the number of ngrams in the training and test sets? How do modifications affect the maximum accuracy rate, and how does it affect the rate of change in the accuracy rate?

3 Methods

3.1 Intuition & Related Work

Several different techniques have been used to perform part-of-speech tagging, including Markov models, rule-based learners, neural networks, and several types of decision tree learners [2, 3, 7, 4]. These algorithms have often been supplemented with techniques such as patching – a process in which a separate holdout set is used to rectify learning mistakes – in order to boost accuracy rates [3, 8]. Accuracy rates using patched versions of these algorithms tend to be in the low to mid nineties, though with additional exception handling modifications they can do much better [4].

Though some past work has indicated that additional training data may not always be incredibly helpful to decision-tree learners, most taggers continue eke out as much accuracy as possible by learning from the entire training set [9, 4]. Work on decision trees in other domains has indicated that tree size increases linearly with additional data even when accuracy rates hardly increase; for these reasons, some researchers indicate that additional training may not be worthwhile after a certain point [9, 10]. Despite these findings an overwhelming majority of papers in the tree tagging literature continue to use the maximal amount of training data possible. Thus, though intuition indicates that additional training examples will improve the accuracy rate, this work will attempt to determine if that rate of improvement is meaningful.

Most past work in part-of-speech classification has focused exclusively on tagging each word in the data set based primarily upon its word value and only secondarily on the previous parts of speech [4]. This has been done because it is well known that classifying a word based solely upon its most-common part of speech in a large (several hundred thousand) training corpus will yield an accuracy classification rate close to 90% [1]. Intuition indicates that increasing the number of subtrees should have a positive impact on known words but can leave the learner vulnerable to misclassifying words not in the training set. This work will examine the number of words classified by a per-word subtree and examine how this affects the accuracy rate in order to measure the above effects.

Lastly, past work with decision trees has focused on ngram parsers with small n values, usually values less than or equal to three [4, 5]. Most of this past research in the part-of-

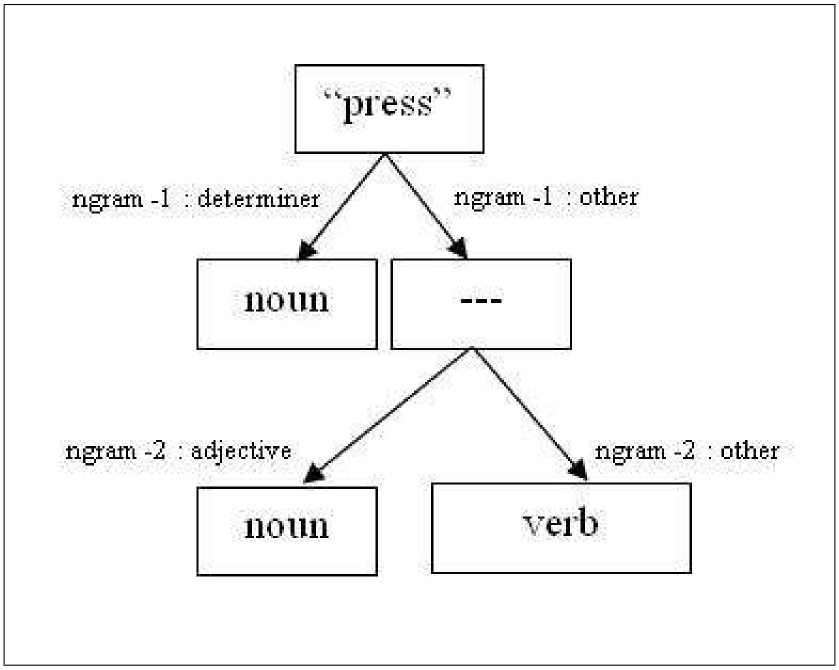

Figure 1: The decision tree for the word "press" as generated by a bigram tagger with a 50,000 example training set

speech tagging domain has focused on unigram, bigram, and trigram taggers, and these taggers have been successful [2, 3]. However, a past experiment using decision trees has indicated that quadrogram trees may be more effective at predicting parts of speech, though the accuracy gain was minimal (less than .05%) as compared to a trigram tagger [4]. Intuitively, it seems as if the greater predictive power of a more complicated decision tree may be able to disambiguate a part of speech tag by providing additional information which the tagger can use to its advantage. This work will evaluate these results and intuitions.

3.2 Data & Algorithms

The part-of-speech decision-tree taggers grown in this project were binary taggers. Binary decision trees were selected as they have been used successfully in previous work on part-of-speech tagging [2]. Binary decision trees, in contrast to other types of decision trees, choose to split based on the information gain that comes from splitting on the particular value of individual attribute at an ngram as opposed to on all attributes of an ngram. The process of binary tree growth is illustrated in figure 1. In figure 1, the sample decision tree grown for the word "press" performs its first split on whether or not the previous part of speech is a determiner: if the previous part of speech is a determiner, then "press" is tagged as a noun; otherwise, other part-of-speech tags are considered.

When growing the binary decision trees, three factors were varied: the size of the training set, the number of word-specific subtrees grown, and the number of previous parts-of-speech considered when making a tagging decision.

- Six different training set sizes were used. These sets had 1000, 5000, 10,000, 25,000, 50,000, and 100,000 labeled examples, though some tests could not be performed for the largest training set size due to performance reasons. Training sets were held consistent across the number of word-specific subtrees grown and the number of ngrams used in order to facilitate comparisons.

- Five different word-management strategies were used. In the simplest case, word-specific data was ignored and part-of-speech prediction was done using only previous parts-of-speech. In the other four cases, words that appeared 400+, 200+, 100+, and 50+ times in the full training corpus were tagged using their own specific subtree. This meant the tagger grew one giant tree in the first case, and approximately 250 word-specific subtrees, 500 word-specific subtrees, 1000 word-specific subtrees, and 2000 word-specific subtrees in the other four respective cases.
- Four different types of tree taggers were grown. The tree taggers were either unigram, bigram, trigram, and quadrogram taggers, meaning that they considered the preceding one, two, three, and four tags when making their tagging decision. Since the binary tagging process meant that each ngram could be split once per part of speech per ngram, each additional ngram greatly increased the number of possible split points by a large amount and caused some computations to become infeasible for large data sets.

All training sets were tested with several independent test sets of 50,000 randomly-generated examples that were not in any of the training sets. To facilitate comparisons, the test sets were controlled so that the same example was used to test all the trees.

4 Experiments

4.1 Testbed

The tree taggers in this project were trained on data from the Penn Treebank corpus of Wall Street Journal articles; this corpus contained about a million words tagged by part-of-speech from which examples were randomly selected [11]. The Treebank data set differentiated between thirty-eight parts of speech: for instance, it draws a distinction between singular and plural nouns as well as nearly a half-dozen types of verbs [12].

Tree training was performed on an IBM T60 computer with a 2.16 GHz processor and 1GB of RAM. The actual machine learning implementation was done using the J48 (C4) decision tree learning package in Weka 3.4 [13]. The time necessary for training varied, but all but the largest decision trees could be grown and pruned in about ten minutes. Due to the size of the training sets and the number of possible words for which to grow trees, however, memory constraints were a major problem. Weka's implementation of decision tree learners made it impossible to grow decision trees for cases with large numbers of examples, as it required large quantities of memory to begin parsing large training sets.

4.2 Results

4.2.1 Increasing the amount of training data

The rows in tables 1 and 2 demonstrate that increasing the size of the training set had a positive impact on training rate. As the number of training examples increased, the accuracy rate rose substantially in most cases. In some cases, however, accuracy rate hardly increased. For instance, both sets of giant trees as well as the unigram tree taggers with 250 or 500 word-specific subtrees failed to improve their accuracy rates significantly after 10,000 training examples. The giant trees appeared to reach an asymptotic maximum near 31%, while the trees with 250- and 500 word-specific subtrees stabilized just shy of 76% and 79% respectively. Though these trees did continue to benefit from the training data, the benefit was limited by the design of the tree.

Table 1: Unigram tagger accuracy rate

TAGGER	1k	5k	10k	25k	50k	100k
One giant tree grown	28.93%	30.64%	30.77%	30.86%	31.06%	N/A
...with 250 word-specific subtrees	69.51%	74.16%	74.77%	75.38%	75.70%	75.87%
...with 500 word-specific subtrees	68.70%	75.74%	77.09%	78.17%	78.42%	78.66%
...with 1000 word-specific subtrees	68.66%	76.75%	78.80%	80.42%	81.12%	81.32%
...with 2000 word-specific subtrees	67.90%	76.75%	79.62%	82.18%	83.49%	N/A

Table 2: Bigram tagger accuracy rate

TAGGER	1k	5k	10k	25k	50k	100k
One giant tree grown	25.94%	27.96%	29.97%	31.05%	31.48%	N/A
...with 250 word-specific subtrees	67.17%	76.29%	79.16%	82.17%	83.58%	N/A
...with 500 word-specific subtrees	67.10%	75.58%	78.27%	82.23%	83.33%	N/A
...with 1000 word-specific subtrees	66.17%	75.61%	79.49%	82.52%	82.73%	N/A
...with 2000 word-specific subtrees	66.28%	74.67%	79.11%	80.69%	83.57%	N/A

Table 3: Accuracy rate for tagger with 250 word-specific subtrees

TAGGER	1k	5k	10k	25k	50k	100k
Unigram tagger	69.51%	74.16%	74.77%	75.38%	75.70%	75.80%
Bigram tagger	66.28%	74.67%	79.11%	80.69%	83.57%	N/A
Trigram tagger	66.54%	75.05%	77.62%	80.85%	82.68%	N/A
Quadrogram tagger	66.56%	74.89%	77.62%	80.64%	82.19%	N/A

Table 4: Accuracy rate for tagger with 1000 word-specific subtrees

TAGGER	1k	5k	10k	25k	50k	100k
Unigram tagger	68.66%	76.75%	78.80%	80.42%	81.12%	81.32%
Bigram tagger	67.10%	75.58%	78.27%	82.23%	83.33%	N/A
Trigram tagger	66.73%	75.14%	78.16%	81.18%	82.75%	N/A
Quadrogram tagger	67.01%	75.16%	77.45%	80.27%	81.89%	N/A

Similarly, the rows of tables 3 and 4 indicate that increasing the size of the training set increased accuracy predictions. As the number of training examples increased, the accuracy rate also increased substantially. However, these tables also indicate that the unigram taggers failed to benefit as much from additional training data; in both tables, there was little increase in accuracy when doubling the number of training examples from 25,000 to 50,000. In contrast to unigram taggers, the taggers that examined additional ngrams continued to improve in this interval.

Taken together, these tables demonstrate that the trees that benefited from additional training data were the more complex decision trees. Often – but not always – these were the trees with the highest accuracy rate when trained on 50,000 examples. Though all the trees did benefit from the additional data, there was a greater impact on the more sophisticated trees. It is also likely that these trees would continue to improve the most with still more data.

4.2.2 Increasing the number of word-specific subtrees

Perhaps the most salient feature of tables 1 and 2 were the shockingly low values recorded in the first row. These values confirm the obvious: knowledge of the word value was necessary and that previous parts-of-speech tags cannot reliably predict subsequent ones. When decisions were made solely based on the parts of speech of the preceding words, as they were for the predictions recorded in the first rows of tables 1 and 2, the prediction rates were slightly worse than 1 in 3. Though this result was better than assigning a random part-of-speech – an accuracy rate slightly higher than 2% – or assigning every test case the statistically most common part of speech – an accuracy rate of roughly 12% for singular noun – this classification scheme is unacceptable for tagging purposes. Thus, while tagging a word based solely on previous parts of speech is more effective than random or educated guessing, it is far from effective ever with large text corpora.

As indicated by the remainder of the columns in table 1, increasing the number of word-specific subtrees had a large positive impact on classification success rates for unigram tree data. Including additional word-specific subtrees caused perceptible increases in accuracy, as doubling the number of words classified using word-specific increased the classification accuracy by between 2% and 3% when using large training sets.

However, the results from table 2 indicate that bigram taggers show little impact from increasing the number of word-specific subtrees. Bigram taggers achieved an accuracy rate of 83.57% when trained using only 250 word-specific subtrees and a similar accuracy rate of 83.58% when trained using 2000 word-specific subtrees using a training corpus of 50,000 examples. Similar results were found for trigram and quadrogram cases: increasing the number of exception trees was of little value for the size of data sets tested. In these cases, providing a few subtrees for classifying the most common words provided an immediate and substantial benefit, though providing additional subtrees for additional words increased the tree complexity but provided little classification benefit.

4.2.3 Increasing the number of ngrams

As the columns of table 3 suggests, unigram tagger accuracy rates were much lower than those for other taggers. This fact was especially apparent when the number of exception words was low, though it was still perceptible even when the number of runs was increased. Additionally, while table 3 may have suggested that unigram accuracy rates are similar to those of the other ngram trees, table 4 and other experiment data indicate exactly the opposite. As discussed in section 4.2.1, unigram taggers received less benefit from additional training data. The data in this table suggested that one could increase accuracy by moving to a bigram or higher type of decision tree.

Tables 3 and 4 indicate that the bigram tree tagger is the superior tree tagger for this sized training set. However, they also suggest that changing between bigram, trigram, and quadrogram taggers had only minimal impact as compared to using a unigram tagger. Even with less training data, the difference in accuracy between bigrams, trigrams, and quadrograms was never much more than about 2%. More training data is necessary to determine whether bigram decision trees are actually superior to their more complicated cousins, though these findings indicate that the additional complexity of the more complicated taggers was not helpful. Though past research has suggested that quadrogram decision trees are superior to their less-complicated counterparts, this could not be confirmed with this part-of-speech tagger [4].

5 Conclusion

This paper examined three possible ways to modify a part-of-speech tagger's training set: increasing the number of training examples, increasing the number of words treated specially in the training data, and increasing the complexity of each example by increasing the number of ngrams. The results of section 4.2.1 indicated that in most cases, increasing the number of examples provided the greatest benefit. However, the results also demonstrated that blindly increasing the number of training examples was not always be effective. The results presented in section 4.2.2 showed that accuracy rates (and rates of change in accuracy rates) could be improved by including at least a small number of word-specific subtrees. Though experimental data here indicated that there was only a slight benefit to increasing the number of word-specific subtrees in most cases, they also suggested that unigram taggers were substantially improved when they were. The results of section 4.2.3 demonstrated that accuracy rates could be improved by examining at least two ngrams (but preferably no more) when training and testing the data. Though the bigram part-of-speech tagger was only slightly better than the trigram or quadrogram one, the data indicated that all three were clearly superior to a unigram tagger.

This paper opens up several avenues for further investigation. Researchers may find it useful to fit functions or curves to the trends identified here. In this manner, they may be able to predict a possible accuracy rate given a set of training set constraints or, conversely, identify a set of constraints on a data set to create a part-of-speech decision-tree tagger with a given accuracy rate. Alternatively, other researchers with access to computers with more memory may be able to expand upon this research by examining cases with more data. By doing this, and evaluating the results on other text corpora to ensure that the results are consistent, they may be able to determine the extent to which the results generalize.

References

[1] E Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44, 1997.

[2] S Abney. Part-of-speech tagging and partial parsing, 1996.

[3] E Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[4] H Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, 1994.

[5] H Schmid. Improvements in part-of-speech tagging with an application to german, 1995.

[6] C Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] D Cutting, J Kupiec, J Pedersen, and P Sibun. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[8] T Brants. Tnt – a statistical part-of-speech tagger, 2000.

[9] T Oates and D Jensen. The effects of training set size on decision tree complexity. In *Proc. 14th International Conference on Machine Learning*, pages 254–262. Morgan Kaufmann, 1997.

[10] T Oates and D Jensen. Large datasets lead to overly complex models: An explanation and a solution. In *Knowledge Discovery and Data Mining*, pages 294–298, 1998.

[11] M Marcus, G Kim, M Marcinkiewicz, R MacIntyre, A Bies, M Ferguson, K Katz, and B Schasberger. The Penn treebank: Annotating predicate argument structure, 1994.

[12] Beatrice Santorini. Part-of-speech tagging guidelines for the penn treebank project (3rd revision, 2nd printing), ms., department of linguistics, upenn. philadelphia, pa.

[13] I Witten, E Frank, L Trigg, M Hall, G Holmes, and S Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.