Performance Analysis of a Part of Speech Tagging Task

Rada Mihalcea

University of North Texas
Computer Science Department
Denton, TX, 76203-1366
rada@cs.unt.edu

Abstract. In this paper, we attempt to make a formal analysis of the performance in automatic part of speech tagging. Lower and upper bounds in tagging precision using existing taggers or their combination are provided. Since we show that with existing taggers, automatic *perfect* tagging is not possible, we offer two solutions for applications requiring very high precision: (1) a solution involving minimum human intervention for a precision of over 98.7%, and (2) a combination of taggers using a memory based learning algorithm that succeeds in reducing the error rate with 11.6% with respect to the best tagger involved.

1 Introduction

Part of speech (POS) tagging is one of the few problems in Natural Language Processing (NLP) that may be considered *almost* solved, in that several solutions have been proposed so far, and were successfully applied in practice. State-of-the-art systems performing POS tagging achieve accuracies of over 93-94%, which may be satisfactory for many NLP applications. However, there are certain applications that require even higher precision, as for example the construction of annotated corpora where the tagging needs to be accurately performed. Two solutions are possible for this type of sensitive applications: (1) manual tagging, which ensures high accuracy, but is highly expensive; and (2) automatic tagging, which may be performed at virtually no cost, but requires means for controlling the quality of the labeling process performed by machine.

POS tagging is required by almost any text processing task, e.g. word sense disambiguation, parsing, logical forms and others. Being one of the first processing steps in any such application, the accuracy of the POS tagger directly impacts the accuracy of any subsequent text processing steps.

We investigate in this paper the current state-of-the-art in POS tagging, derive theoretical lower and upper bounds for the accuracy of individual systems or combinations of these systems, and show that with existing taggers *perfect* POS tagging is not possible (where *perfect* tagging is considered to be 100% accuracy with respect to manually annotated data). Subsequently, we provide two possible solutions for this problem. First, we show that it is possible to design

a tagging scheme that guarantees a precision of over 98.7% with minimum human intervention. Secondly, we show how individual taggers can be combined into a new tagger, with an error reduction of 11.6% with respect to the best tagger involved.

1.1 Classifiers Combination

Combining classifiers for improved performance is a technique well known in the Machine Learning (ML) community. Previous work in the field has demonstrated that a combined classifier often outperforms all individual systems involved [1].

Classifier combination has been successfully used in several NLP problems, including POS tagging [4,14], word sense disambiguation [6,8], and others. Brill [4] and van Halteren [14] show how several POS taggers can be combined using various approaches: voting schemes (simple or weighted), decision trees, and rules learned using contextual cues. Brill [4] uses four taggers: an unigram tagger, an N-gram tagger, Brill's tagger and the Maximum Entropy tagger, for an error reduction of 10.4% with respect to the best tagger involved. In concurrent work, van Halteren [14] combines again four different taggers and obtains a reduction of 19.1% in error rate using a pairwise voting scheme.

In this paper, we attempt to formalize the combination of various taggers for improved accuracy: we provide lower and upper bounds for POS tagging precision. Since we prove that the performance of existing taggers or their combinations cannot exceed a certain limit, we suggest two possible solutions for applications requiring high tagging accuracy: (1) a solution that involves minimum human intervention for an accuracy of over 98.7%; and (2) a combination of taggers using a memory based learning algorithm that succeeds in reducing the error rate by 11.6% with respect to the best tagger involved.

2 Mathematical Foundations

This section describes a mathematical model for the problem of text tagging, and shows how the level of confidence in tagging precision can be formally estimated. First, given the fact that voting is a widely used scheme in classifier combination, we are interested in finding lower and upper bounds for the tagging precision on the set where two taggers agree. Results pertaining to this problem were previously reported in [11]. Additionally, we want to determine lower and upper bounds for the precision on the entire tagged set (including both agreement and disagreement sets).

2.1 Precision on Agreement Set

It was previously shown [11] that given two classifiers with their estimated precisions, it is possible to determine a minimum and a maximum for the precision achieved on the agreement set, i.e. the set where the two classifiers agree in the tag they independently assign.

The following formulae were derived:

$$\min P_{\mathcal{A}_{12}} = \frac{P_{T_1} + P_{T_2} - 1 + \mathcal{A}_{12}}{2 * \mathcal{A}_{12}} \quad (1)$$

$$\max P_{\mathcal{A}_{12}} \simeq \frac{P_{T_1} + P_{T_2} - 1 + \mathcal{A}_{12} + (1 - P_{T_1})(1 - P_{T_2})}{2 * \mathcal{A}_{12}} \quad (2)$$

where P_{T_1} is the precision of the tagger T_1 , P_{T_2} is the precision of the tagger T_2 , $P_{\mathcal{A}_{12}}$ is the precision on the agreement set, $\mathcal{A}_{12}$ is the size of the agreement set. Experiments with POS tagging have validated this result, leading to almost identical theoretical and empirical values.

2.2 Lower and Upper Bounds for Overall Precision

The second problem that we address from a theoretical perspective regards the limitations in POS tagging precision. Similar theoretical analyses were previously performed for the problem of word sense disambiguation [7], and recently for the problem of question answering [9].

The simplest approach in POS tagging is the unigram tagger that lexically disambiguates words based on their frequency in a large corpus, and was found to perform very well, with an accuracy of about 93.26% precision [4]. This can be considered as a **lower bound** (i.e. a *baseline*) for the problem of POS tagging in general.

Finding the upper bound is a process significantly more difficult, since it has to take into account the precision of all individual and combined classifiers. It is however an important issue, since accurate predictions of upper bound would enable a complete analysis of the performance in POS tagging. Moreover, such theoretical evaluations may influence the decision in selecting the best individual / combined tagger for particular applications.

To the end of finding this upper bound, we differentiate the *absolute precision* of a tagger, as compared to the *nominal precision*. It is common to report precision in POS tagging by referring to all words in a particular test set. This is the *nominal precision*. However, a fairly large set of words in any text have only one possible tag. These words are not lexically ambiguous, and therefore they account for a subset with 100% tagging precision towards the overall *nominal precision* of a tagger. The *absolute precision* is the precision achieved by the same tagger when applied only to the set of ambiguous words. Measurements performed on a large tagged corpus have shown that about 40% of the words in the corpus are not lexically ambiguous. This is in agreement with the corresponding figures reported in [4].

By denoting with P_{T_i} the *nominal precision* of a tagger T_i , and with AP_{T_i} the *absolute precision* of the same tagger, the following equation can be written:

$$P_{T_i} = 0.40 * 1.0 + 0.60 * AP_{T_i} \Rightarrow AP_{T_i} = \frac{(P_{T_i} - 0.40)}{0.60} \quad (3)$$

meaning that the nominal precision of a tagger on any set can be divided in two terms: the precision of 100% achieved on 40% of the set (i.e. the set of non ambiguous words), and the absolute precision achieved on the rest of 60% of the set (i.e. the set of ambiguous words).

On the other hand, given two taggers T_1 and T_2 , the overall precision for these two taggers can be written as the precision on the set where the two taggers agree, plus the precision on the set where the two taggers disagree:

$$P = \mathcal{A}_{12} * P_{\mathcal{A}_{12}} + (1 - \mathcal{A}_{12}) * P_{1 - \mathcal{A}_{12}} \quad (4)$$

The maximum overall precision that can be achieved by these two taggers, individually or combined, is determined as the sum of (a) the maximum precision achieved by the two taggers on the agreement set and (b) the maximum *absolute precision* of the individual classifiers on the disagreement set. This is based on the observation that the words that are not lexically ambiguous are all included in the agreement set, and this is why the maximum accuracy that can be achieved by any individual tagger on the disagreement set is given by its *absolute precision*.

$$\begin{aligned} \max P &= \mathcal{A}_{12} * \max P_{\mathcal{A}_{12}} + (1 - \mathcal{A}_{12}) * \max(AP_{T_1}, AP_{T_2}) \\ &= \mathcal{A}_{12} * \frac{P_{T_1} + P_{T_2} - 1 + \mathcal{A}_{12} + (1 - P_{T_1})(1 - P_{T_2})}{2 * \mathcal{A}_{12}} \\ &\quad + (1 - \mathcal{A}_{12}) * \max(AP_{T_1}, AP_{T_2}) \end{aligned} \quad (5)$$

Using this theoretical model, we are able to derive lower and upper bounds in tagging precision, for individual or combined classifiers. In addition to POS tagging, this model is applicable to the analysis of other NLP labeling tasks, such as word sense disambiguation, prepositional attachment and others.

3 Empirical Results

We provide in this section empirical support for the model derived above. First, we measure the performance of four part of speech taggers on a test set extracted from the Penn Treebank corpus. Next, we show that the values empirically determined in practice are very close to the values found in theory. Finally, we apply the formula derived in the previous section to derive an upper bound for the overall precision that can be achieved with these state-of-the-art taggers.

3.1 State of the Art in POS Tagging

Several methods have been proposed so far for POS tagging, including transformation based systems, taggers based on maximum entropy models, or derived using decision trees. The accuracies achieved by these taggers range from 94 to 96%, depending on the method employed, and/or on the training and testing sets.

Transformation based tagger [T1]. The tagger developed by Brill [3] works by first assigning the most frequent tag to each word; next, rules are applied to change the tag of the word based on the context in which they appear. It is one of the most popular taggers, due to its accuracy and public availability.

Maximum entropy tagger [T2]. Mxpost tagger was developed by Ratnaparkhi [12]; it is a statistical model based on maximum entropy, integrating many contextual features.

TnT tagger [T3]. The TnT (Trigrams'n'Tags) tagger is a statistical part of speech tagger, written by Brants [2]; TnT incorporates several smoothing methods, and is optimized for training on a variety of corpora. The tagger is an implementation of the Viterbi algorithm for second order Markov models.

TreeTagger [T4]. TreeTagger [13] is a probabilistic tagger that attempts to avoid the problems encountered by Markov model taggers: the transition probabilities in TreeTagger are estimated using decision trees.

3.2 Experiments

To evaluate the precision of the individual taggers and to determine the upper bound for overall combined precision, the following experiment was performed: the Penn Treebank corpus [10] was divided in two parts: sections 0-19, used for training the four POS taggers, and sections 20-60, separately tagged with each of the four classifiers. Subsequently, during the experiments reported in section 4.2, this second set was divided into a subset of 1,500,000 words for training the classifier combination model, respectively 298,000 words for testing.

The precision of each individual tagger was measured on this last set of 298,000 words. This set is sufficiently large to provide accurate measures of precision, and at the same time enables a fair comparison with the precision of the combined taggers, which are tested on the same subset.

Table 1 lists the precisions for individual taggers, as found empirically on the test set, and the size of the agreement sets and precision achieved on these sets. Using equations 1 and 2, the theoretical values can be computed. Notice that the values associated with the combination of all four taggers are determined in a recursive manner: taggers T_1 and T_2 , respectively T_3 and T_4 are paired and form two "new" taggers, T_{12} and T_{34} , with their associated minimum and maximum precisions. As shown by the results in Table 1, the values determined empirically are tightly close to the figures that can be computed in theory.

Furthermore, with the model described in the previous section, we can induce upper bound values for the precision that may be achieved with various classifiers. Table 2 shows the values determined using equation 5. It follows that the taggers considered in this experiment, individually or combined, cannot exceed the overall precision of 98.43%.

The equation for the maximum precision has an intuitive explanation: it depends on the precision of the individual taggers, and it also depends on the size of their agreement set. Brill [4] has noticed that higher differences among classifiers can lead to higher combined precision. Here, a smaller agreement set (e.g. the agreement set of taggers T_1 and T_2) results in larger overall precision, compared with the smaller overall precision achieved with larger agreement sets (e.g. the agreement set between T_3 and T_4).

4 Solutions for High Precision POS Tagging

In the previous section, we have shown – using theoretical and empirical means – that the precision of current state-of-the-art taggers cannot exceed a certain

Table 1. Values determined empirically on the Penn Treebank corpus

Measure	Notation	Value
Precision of the taggers		
T_1	P_{T_1}	0.9403
T_2	P_{T_2}	0.9598
T_3	P_{T_3}	0.9602
T_4	P_{T_4}	0.9599
Absolute precision of the taggers		
T_1	AP_{T_1}	0.9005
T_2	AP_{T_2}	0.9330
T_3	AP_{T_3}	0.9336
T_4	AP_{T_4}	0.9331
Size of agreement set between taggers		
T_1 and T_2	$\mathcal{A}_{12}$	0.9369
T_3 and T_4	$\mathcal{A}_{34}$	0.9799
T_1, T_2, T_3 and T_4	$\mathcal{A}_{1234}$	0.9155
Precision on agreement set		
T_1 and T_2	$P_{\mathcal{A}_{12}}$	0.9810
T_3 and T_4	$P_{\mathcal{A}_{34}}$	0.9702
T_1, T_2, T_3 and T_4	$P_{\mathcal{A}_{1234}}$	0.9860

Table 2. Maximum overall precisions

Measure	Notation	Value
Maximum overall precision		
T_1 and T_2	$\max P_{12}$	0.9785
T_3 and T_4	$\max P_{34}$	0.9686
Absolute maximum overall precision		
T_1 and T_2	$\max AP_{12}$	0.9641
T_3 and T_4	$\max AP_{34}$	0.9476
Maximum overall precision		
T_1, T_2, T_3 and T_4	$\max P_{1234}$	0.9843

upper limit. In this section, we propose two possible solutions that can be used for sensitive applications where high precision POS tagging is critical.

4.1 Solution 1: Highly Accurate Tagging Using Minimum Human Intervention

One first solution is to use minimum human intervention for an overall higher performance. Given several automatic taggers, we can devise a scheme where a human checks only the part of the corpus where the taggers disagree. Assigning a 100% accuracy to manual tagging, we can determine the minimum overall precision achieved by various combinations of taggers, using a variation of equation

4, which encodes the precision of 100% achieved by a human on the disagreement set:

$$\min P_{12h} = \mathcal{A}_{12} * \min P_{\mathcal{A}_{12}} + (1 - \mathcal{A}_{12}) * 1 \quad (6)$$

Table 3 shows the minimum values for overall precision for various combinations of taggers, and the size of the set that has to be manually checked.

Table 3. Minimum overall precisions achieved with human intervention

Measure	Notation	Value
Minimum overall precision		
T_1 and T_2	$\min P_{12h}$	0.9815
T_3 and T_4	$\min P_{34h}$	0.9700
T_2 and T_3	$\min P_{23h}$	0.9717
T_1, T_2, T_3 and T_4	$\min P_{1234h}$	0.9871
Disagreement set (to be manually checked)		
T_1 and T_2	$1 - \mathcal{A}_{12}$	0.0631
T_3 and T_4	$1 - \mathcal{A}_{34}$	0.0201
T_2 and T_3	$1 - \mathcal{A}_{23}$	0.0349
T_1, T_2, T_3 and T_4	$1 - \mathcal{A}_{1234}$	0.0845

It is debatable what combinations are better from the point of view of precision and recall. A compromise between precision, recall and number of taggers involved is achieved using the first two taggers. It guarantees a minimum precision of 98.15%, significantly larger than the best tagger involved, with 6% of the tags being checked by a human.

4.2 Solution 2: Combining Taggers for Improved Precision

An alternative solution is to combine different classifiers using a machine learning approach. The combination of taggers reported in [4] led to an error reduction of 10.3%; we follow their direction and show that a memory based learner can lead to a slightly higher reduction in error rate, of 11.6%.

The learning system is Timbl [5]. It is a memory based learner that works by storing training examples; new examples are classified by identifying the closest instance in the training data. The similarity among examples is computed using an overlap metric (Manhattan metric), improved with an information gain feature weighting. It provides good results in short time (learning from about 125,000 examples and testing 25,000 examples takes about 9 seconds).

First, we tagged sections 20-60 from Treebank using all four part of speech taggers. Next, we divided this corpus in two sets: a training set, comprising 1,500,000 words and a testing set with 298,000 words. Finally, we eliminated those examples for which all taggers agree. There are two reasons for this decision: first, we have already proved that very high precision (98.6%) can be

achieved on the agreement set, and therefore we want to focus only on the remaining “problematic” cases where the taggers disagree. Second, the cases where all taggers agree account for about 91% of the examples. Since we use a learning algorithm that computes the difference between training examples and testing examples, these large number of cases where all taggers agree will favor the majority voting algorithm, which might not always be the best decision.

After eliminating these cases when $T_1=T_2=T_3=T_4$, we are left with a training set of 126,927 examples, and a testing set of 25,248 examples; they both contain only disagreement examples. The precisions of the four taggers on these test examples are 40.19%, 63.24%, 63.70% and 63.44%.

The following sets of features are used for learning:

- **4T**. The tags assigned by the four taggers.
- **4T+TB**. The tags assigned by the four taggers and the tag of the word before the current word (assigned by the best performing tagger, i.e. TnT)
- **4T+TB+TA**. The tags assigned by the four taggers, plus the tag of the word before, plus the tag of the word following the current word.
- **4T+TB+W**. The tags assigned by the four taggers, plus the tag of the word before, plus the word itself.
- **4T+TB+TA+W**. The tags assigned by the four taggers, plus the tag of the word before, plus the tag of the word after, plus the word itself.

Table 4 shows the precision achieved using the memory based learning algorithm on the test set. We also compute the minimum overall precision, using equation 4.

Table 4. Precision for the combination of four taggers, for various sets of features.

Feature set	Precision on test set (disagreement set)	Min.overall precision
4T	70.10%	96.19%
4T+TB	71.06%	96.27%
4T+TB+TA	71.59%	96.31%
4T+TB+W	72.11%	96.36%
4T+TB+TA+W	73.55%	96.48%

Using the last set of features, which includes the tags assigned by the four taggers and contextual clues, the largest reduction in error rate is obtained. On our test set, the reduction in error rate is 27% (73.55% vs. the best tagger precision of 63.70% on the same set). For the overall precision, the reduction in error rate is 11.6% with respect to the best tagger involved.

5 Conclusion

We have addressed in this paper the limitations of existing POS taggers. Even though current state-of-the-art systems provide high accuracies in the range of 94-96%, we have shown that the precision of individual or combined taggers cannot exceed an upper bound of 98.43%. Two solutions have been provided for sensitive applications requiring highly precise tagging. First, we have shown how minimum human intervention can guarantee a minimum overall precision of 98.7%. Second, we have shown that a combination of existing taggers using a memory based learning algorithm succeeds in reducing the error rate with 11.6% with respect to the best tagger involved.

We have also derived a theoretical model for the analysis of lower and upper bounds in POS tagging performance. This theoretical scheme is equally applicable to other NLP labeling tasks such as word sense disambiguation, prepositional attachment and others.

References

1. ALI, K., AND PAZZANI, M. Error reduction through learning multiple descriptions. *Machine Learning* 24, 3 (1996), 173-202.
2. BRANTS, T. Tnt - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000* (Seattle, WA, May 2000).
3. BRILL, E. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics* 21, 4 (December 1995), 543-566.
4. BRILL, E., AND WU, J. Classifier combination for improved lexical disambiguation. In *In Proceedings of the Seventeenth International Conference on Computational Linguistics COLING-ACL '98* (Montreal, Canada, 1998).
5. DAELEMANS, W., ZAVREL, J., VAN DER SLOOT, K., AND VAN DEN BOSCH, A. Timbl: Tilburg memory based learner, version 4.0, reference guide. Tech. rep., University of Antwerp, 2001.
6. FLORIAN, R., CUCERZAN, S., SCHAFER, C., AND YAROWSKY, D. Combining classifiers for word sense disambiguation. *JNLE Special Issue on Evaluating Word Sense Disambiguation Systems* (2002). forthcoming.
7. GALE, W., CHURCH, K., AND YAROWSKY, D. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)* (1992).
8. KLEIN, D., TOUTANOVA, K., ILHAN, I., KAMVAR, S., AND MANNING, C. Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the ACL Workshop on "Word Sense Disambiguation: Recent Successes and Future Directions* (July 2002), pp. 74-80.
9. LIGHT, M., MANN, G., RILOFF, E., AND BRECK, E. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering (forthcoming)* (2002).
10. MARCUS, M., SANTORINI, B., AND MARCINKIEWICZ, M. Building a large annotated corpus of english: the Penn Treebank. *Computational Linguistics* 19, 2 (1993), 313-330.

11. MIHALCEA, R., AND BUNESCU, R. Levels of confidence in building a tagged corpus. Technical Report, SMU, 2000.
12. RATNAPARKHI, A. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference* (Philadelphia, May 1996), pp. 130–142.
13. SCHMID, H. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing* (Manchester, UK, 1994).
14. VAN HALTEREN, H., ZAVREL, J., AND DAELEMANS, W. Improving data driven wordclass tagging by system combination. In *In Proceedings of the Seventeenth International Conference on Computational Linguistics (COLING-ACL '98)* (Montreal, Canada, 1998).