Selective Classifiers for Part-of-Speech Tagging

Erin Renshaw
Christopher J.C. Burges
Ran Gilad-Bachrach
One Microsoft Way
Redmond, WA 98052

MSR Technical Report MSR-TR-2014-64
May 16, 2014

Abstract

We investigate the use of *selective classifiers* for part-of-speech tagging (POS). The idea is to allow classifiers to abstain on hard instances, passing them to downstream classifiers that may have more context available. In this report we focus on just the first stage of such a cascade, and ask whether selective classifiers attain the accuracies needed on those instances they accept, given that such instances will not be revisited by downstream processing. We show that a selective classifier that is constructed as an *abstaining committee* of two off-the-shelf POS taggers can indeed achieve very high accuracies with modest drops in coverage. We also compute the overall accuracy when all instances are voted on by applying majority vote to the abstentions, and we find that this results in state of the art accuracies, robustly.

1 Introduction

There is a substantial amount of work on propagating uncertainty through the NLP pipeline. For example, Finkel et al. (2006) model the NLP pipeline as a Bayesian network and propose a simple and effective sampling technique to perform approximate joint inference. Auli and Lopez (2011) investigate using both loopy belief propagation and dual decomposition to combine supertagging and parsing (replacing their use in a pipeline) for a Combinatory Categorial Grammar parser. However, in general, approaches that rely on loopy graphical models are limited by the need to avoid the intractability of inference in such models (Koller and Friedman, 2009). If instead we take the view that such a pipeline consists of modules where more context becomes available as the processing moves downstream, the question naturally arises as to whether the problem could be approached by modeling the decisions using a sequence of finite state machines, rather than treating the problem as a single inference problem. In this report we investigate a preliminary step towards a simple method for building modular text processing systems, where module outputs can be updated easily based on feedback from downstream modules for which more textual context is available. We use *Selective Classifiers* (El-Yaniv and Wiener, 2010). A selective binary classifier is simply a classifier with three outputs: the hypothesized class $\{\pm 1\}$, and *abstain*, denoted by '$*$' below. The central question addressed in this work is: if we apply selective classifiers to a fundamental natural language problem such as part of speech (POS) tagging, can accuracy rates on the accepted instances, together with coverage (the fraction of accepted instances), be made high enough that applying such methods to solve the feedback problem is even feasible? In addition, we demonstrate a simple combination technique using selective classifiers that robustly gives state of the art results on POS tagging, which gives some indication that this approach may be worth pursuing.

We emphasize that selective classifiers have further advantages beyond the main focus of this study, namely: if such classifiers are linked in a cascade in such a way that the bulk of the data is classified by the first few classifiers in the cascade, then models that would be prohibitively slow to use on all the data can be applied to small, targeted subsets that are abstained on by the upstream, simpler classifiers. Similar ideas have been investigated for object detection in images

(Viola and Jones , 2001). Second, by breaking the problem into components, the overall system can be made more interpretable, since errors made by classifiers that are specialized to specific tasks are often easier to understand than errors made by a single general purpose classifier. Finally, the compartmentalization would also makes the overall system more correctable (that is, its overall decision surface is more stable when adding training data to correct errors), since only the classifier making the error need be corrected. However, we are getting ahead of ourselves: in this paper, we focus on just the first stage of such a cascade, and we ask the preliminary question of whether or not the accuracy/coverage tradeoff there will be good enough to support such a model.

2 The Data

We used the standard benchmark set for POS tagging, namely the Wall Street Journal (WSJ) data from the Penn TreeBank v3 (Marcus et al., 1993), sections 0-18 for training and 22-24 for testing (we did not use the development set, sections 19-21). The WSJ data has noisy labels. For example, on the training set, for the phrase “Chief Executive Officer”, the token “Executive” is labeled as either a noun or an adjective in roughly equal numbers. In order for our simple selective classification scheme to be useful, it is important that error rates on the accepted instances be very low, since such errors are not revisited. To attain a clearer picture of the value of this approach we thus needed a dataset with more reliable labels. To this end, we used *MC160*, a set of 160 short stories gathered using crowdsourcing (Richardson et al., 2013). We chose this data because the vocabulary is limited to that of a typical seven year old, thus limiting the occurrence of labeling ambiguities, yet the data is also open domain. We labeled all tokens as either noun, or not noun, using the Oxford English Dictionary as arbiter, with one exception: we noticed that the OED labels indefinite pronouns as pronouns, whereas they are consistently labeled as nouns in the WSJ corpus; for overall consistency we therefore labeled all indefinite pronouns (namely, *someone, somebody, something, anyone, anybody, anything, everyone, everybody, everything*) as nouns. For this data, we used the first 100 stories as the training set and the remaining 60 as the test set.[1]

3 Related Work

The state of the art for POS tagging accuracies (all tokens) on the WSJ data is between 97.0% and 97.5%. The ACL Wiki on the POS tagging state-of-the-art lists 12 systems whose accuracies lie in this range (ACL Wiki, 2014).

3.1 Selective Classifiers

El-Yaniv and Wiener (2010) analyze the properties of selected classifiers in the noise-free case, that is, the case where the data is separable. A *selective classifier* is a pair f, g with $f: x \to \{\pm 1\}$ and $g: x \to [0,1]$ such that

$$(f,g)(x) \mapsto \begin{cases} \text{reject w.p. } 1 - g(x) \\ \quad f(x) \text{ otherwise.} \end{cases}$$

The *risk* is then defined as the expected loss on the accepted samples, and the *coverage* as $\mathbb{E}[g(X)]$. As motivation for our work, note that in the case in which the function class is both finite, and contains the target function f^*, zero risk can in fact be achieved with guaranteed finite coverage using the *Consistent Selective Strategy* (CSS), in which case the coverage is bounded below by

$$1 - \frac{1}{m} O\left(|\mathcal{F}| + \ln\left(\frac{1}{\delta}\right)\right)$$

with probability $1 - \delta$. Here, m is the number of IID training samples and $|\mathcal{F}|$ the size of the function class. CSS is very simply implemented by using the training data to select that subset of functions that predict it perfectly; by assumption, f^* is in that set; and a given test point is accepted only if all such functions agree on it, otherwise CSS abstains, guaranteeing that the assigned class is also that assigned by f^* (El-Yaniv and Wiener, 2010).

3.1.1 Cascades of Selective Classifiers

One can build a cascade out of a sequence of selective classifiers, such that the second stage classifier attempts to classify all samples on which the first stage classifier abstains, the third does the same with the second’s abstentions, and so on. Although in this paper we only consider a single stage, our hope is to help lay the groundwork for using cascades for text processing, and so we note here a simple generalization bound for such cascades. Throughout the paper, we

[1] These labels will be made available at http://research.microsoft.com/mct.

will only consider the simple case where g is deterministic, i.e. $g: x \rightarrow \{0,1\}$. Let $x \in X$ and let μ denote the measure on $X \times \{\pm 1\}$. We define an ϵ, ρ selective classifier $c: X \rightarrow \{\pm 1, *\}$ to be one for which $P_{x,y\sim\mu}(c(x) \in \{\pm 1\}) \geq \rho$ and $P_{x,y\sim\mu}(c(x) \neq y | c(x) \in \{\pm 1\}) \leq \epsilon$, that is, whose coverage is at least ρ and whose risk is at most ϵ. Then the following

Algorithm: *Repeatedly train new $\epsilon,\ \rho$ selective classifiers on the set of examples for which all previous classifiers abstained, until the probability mass of the remaining abstentions is $\leq \epsilon$*

satisfies the following

Theorem: The resulting hypothesis has generalization error $\leq 2\epsilon$, and the algorithm will run in at most $\frac{\log \epsilon}{\log(1-\rho)}$ iterations.

Proof: Label the regions on which each classifier outputs $\{\pm 1\}$ as $X_1, \ldots X_n$, and let X_{n+1} denote the final abstention region. Then $P(error) = \sum_{i=1}^{n+1} P(error|X_i)P(X_i) \leq \sum_{i=1}^{n} \epsilon\rho_i + \rho_{n+1}.1 \leq 2\epsilon$ since $\sum_{i=1}^{n+1} \rho_i = 1$ and $\rho_{n+1} \leq \epsilon$. Also after the ith iteration, the abstention mass is at most $(1-\rho)^i$; hence we stop for that i for which $(1-\rho)^i \leq \epsilon$ or $i \geq \frac{\log \epsilon}{\log(1-\rho)}$.

□

4 The Abstaining Committee

Our (first stage) model, which we call an *Abstaining Committee,* is built using two "black box" component models: SPLAT and NLPLib. SPLAT is a publicly available language analysis toolkit (Quirk et al., 2012). We used SPLAT's POS tagger and constituency tree parser to provide two sources for POS hypotheses. SPLAT's POS tagger is a maximum entropy Markov model trained on POS tags from the Penn TreeBank (Marcus et al., 1993). Its constituency parser is trained on the Wall Street Journal portion of the Penn TreeBank.

NLPLib (Chen, 2012) is an NLP toolkit that uses the averaged perceptron algorithm (Collins, 2002) trained on the POS and constituency tree tags and data in OntoNotes Release 4.0 (Weischedel et al., 2010). As for SPLAT, we used the NLPLib POS tagger and constituency tree outputs to provide two sources for POS hypotheses. In the following, we refer to these four POS taggers as the base taggers.

Using these four component models, we built a composite model as follows. First, the base taggers were run on the data, mapping each token to a 4-vector with components indexed by the four POS taggers, and containing either a POS tag or "abstain". Then, a "voting table" is formed where each row corresponds to one particular 4-vector. Thus, for example, if one row corresponds to the vector $[NN, NN, JJ, *]$, then two of the base taggers voted n*oun, singular or mass,* one voted *adjective,* and one abstained. In the worst possible case, the number of rows would be $n_{max} = 43^4$, since there are 42 different tags in the Penn TreeBank. However if all base taggers agreed and none abstained, then the total number of rows would be the number of different parts of speech encountered in the text, which is at most 42. Indeed, the number of rows is typically much smaller than n_{max} since the disagreements are rare and tend to occur in patterns. At this stage abstentions only occur when SPLAT and NLPLib disagree on the tokenization (we used the SPLAT tokenization, so the SPLAT base taggers never abstain, and the NLPLib base taggers abstain only when they encounter a token that does not occur in NLPLib's tokenization). Finally, two more columns are added to the table as follows: the first contains the fraction of the training set that is correctly classified, for whichever POS is correctly classified the most, and the second contains that POS tag. For example, if whenever the base classifiers vote $[NN, NN, JJ, *]$, the measured frequency of NN being the correct tag is X, similarly Y for JJ, and Z for, say, *VB*, so that $X + Y + Z = 1$, and if $X = \max(X, Y, Z)$, then the fifth column in the table would be X, and the sixth, *NN*. Finally, we also introduce an accuracy threshold θ, the only parameter in our model. Denoting the voting table by T, then for a given θ, whenever $T_{i5} \geq \theta$, then the abstaining committee outputs T_{i6}, else it abstains. In this way we are guaranteed that for those instances for which the combined classifier does not abstain, the accuracy on the training set is bounded below by θ.

5 Results

The first four rows of Tables 1 and 2 show the accuracies of the base classifiers on the WSJ and MC160 test data sets, respectively. The fifth row of Table 1 shows the results of a majority vote (type I) where ties are broken by taking the highest frequency POS in the training set; thus, for example, if the 4-vector were *[NN, NN, JJ, JJ]*, and more tokens were labeled *NN* than were labeled *JJ* in the entire training set, then the output hypothesis would be *NN*. The fifth row in Table 2 is the analog for nouns only (i.e. the vote is “not noun”, since the majority of tokens in the train set are not labeled as nouns). The sixth row of both tables shows the results of an alternative majority vote (type II) where the hypothesis of that base tagger with the highest accuracy on the training set is used. The seventh row shows the result of choosing θ, the minimum accuracy on the training set, to be 0.98. Thus, by abstaining on 6.5% of the WSJ data, we are able to achieve 99.2% accuracy on the accepted data. The final rows shows the results of forcing the abstaining committee to commit by simply performing majority vote on the abstentions, using each type of majority vote. Note that on the WSJ data, the NLPLib POS tagger performs quite poorly, but we nevertheless find that combining these four systems into an abstaining committee gives state of the art results, which suggests that abstaining committees are quite robust. It is also striking that this strategy does considerably better than simply performing majority vote I on all instances, showing further robustness to the kind of majority voting used on abstentions. The MC160 test data shows that, in the case where the labels are clean and the problem simpler, the abstaining committee achieves close to 100% accuracy on the accepts: by dropping the coverage from 100% to 97.1%, the accuracy improves by 75% relative (from 99.2% to 99.8%). The simplicity of the MC160 noun detection task as compared to the WSJ POS task is also indicated by the number of rows in their corresponding voting tables, which are 17 and 2,712 respectively.

Figure 1 shows the dependency of the train and test accuracies, and the coverage, on θ, for the WSJ test data. Figures 2, 3 and 4 show the same for accuracy, precision and recall for the MC160 data. It is striking that the performances consistently exceed that of the best single system (the best single system performance, and worst single system performance, are denoted by horizontal lines in the figures), demonstrating a “wisdom of the crowd” effect. The left *y* axis uses log base 10 so that, for example, 99.99% accuracy maps to 3, 99.9% to 2, and so on. The right *y* axis applies only to the two coverage curves.

System	Accuracy/%
SPLAT POS	96.4
SPLAT constituency tree	96.7
NLPLib POS	85.0
NLPLib constituency tree	94.5
Majority vote I	96.0
Majority vote II	97.5
Component model, θ =0.98, 93.5% coverage	99.2
Component model with majority vote I θ = 0.98	97.5
Component model with majority vote II θ = 0.98	97.5

Table 1: Results on the WSJ test Data

System	Accuracy/%
SPLAT POS	98.4
SPLAT constituency tree	98.7
NLPLib	98.6
NLPLib constituency tree	99.1
Majority vote I	99.2
Majority vote II	99.1
Component model, θ = 0.98, 97.1% coverage	99.8
Component model with majority vote I θ =0.98	99.2
Component model with majority vote II θ=0.98	99.1

Table 2: Results on the MC160 test data

6 Conclusions

We have shown that abstaining committees built from off-the-shelf POS taggers can produce very high accuracy results with modest loss in coverage. Further, we found that applying majority vote to the abstentions results in state of the art accuracies, and that these results were robust to two choices of how the majority vote broke ties. This suggests that, when building a cascade using these ideas, the majority vote results could also be used as inputs to downstream classifiers to help provide a strong initial baseline. Selective classifiers have other advantages in terms of efficiency (since the bulk of the data may be handled by simple, fast classifiers) and interpretability and correctability (since each classifier works on a subtask of the overall problem). Given these results, a natural next step would be to build on these ideas and investigate using cascades of abstaining committees for natural language tasks.

Acknowledgments

We thanks Aitao Chen for providing NLPLib. We thank Chris Quirk, Lucy Vanderwende and Kristina Toutanova for help with SPLAT and for useful discussions.

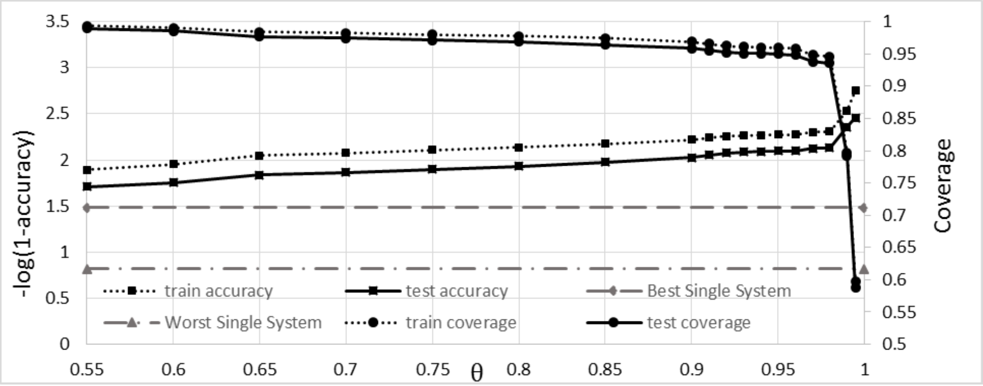
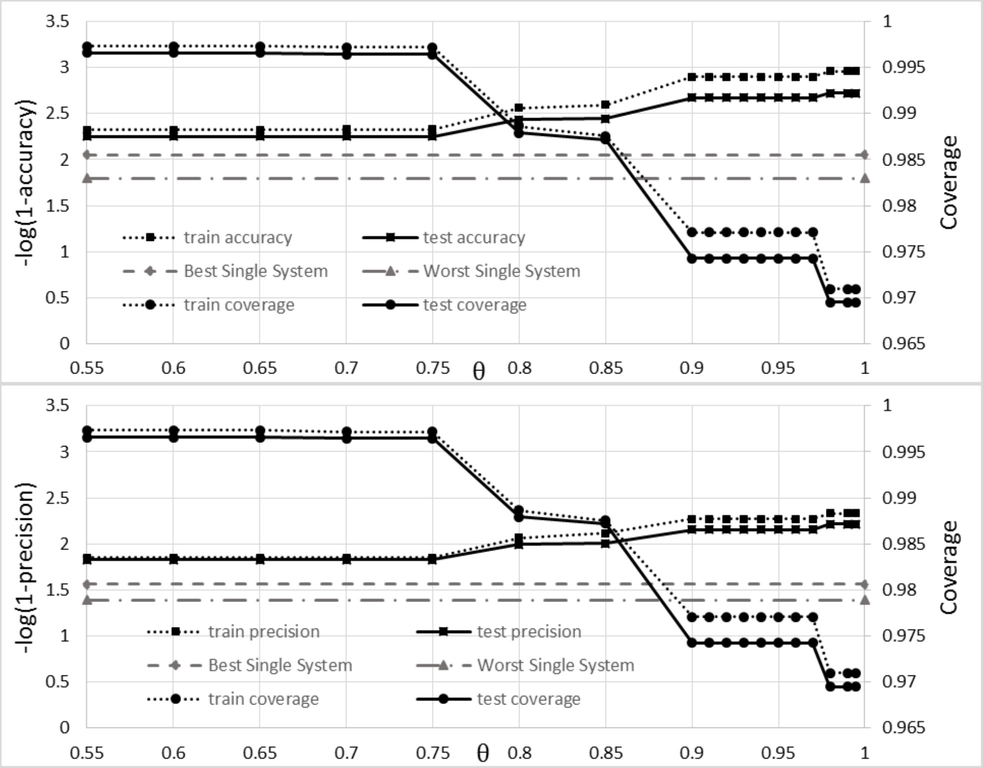


Figure 1 Accuracy vs. threshold for the WSJ train and test sets. The coverage curves drop from left to right while the accuracy curves are flat or increase (as for Figure 2).



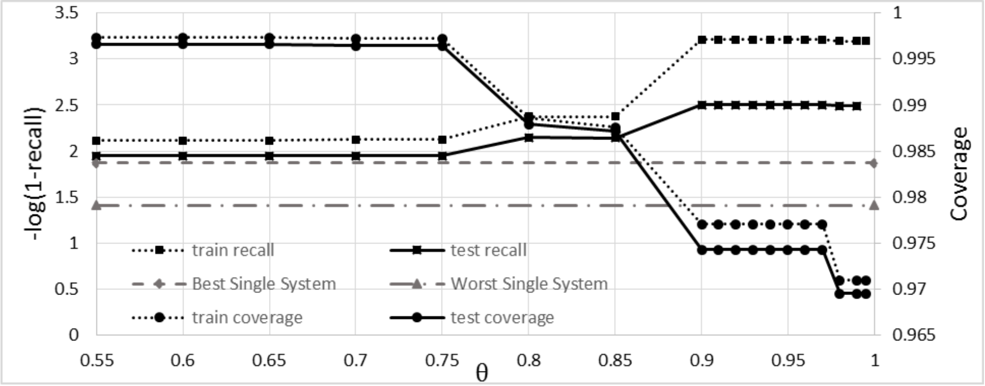


Figure 2: Accuracy, precision and recall vs. threshold for the MC160 train and test sets.

References

ACL Wiki on the State of the Art on Part of Speech Tagging.2014.*http://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)*. Association for Computational Linguistics.

Michael Auli, and Adam Lopez. 2011. *A comparison of loopy belief propagation and dual decomposition for integrated CCG supertagging and parsing.* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 470-480. Association for Computational Linguistics.

Michael Collins. 2002. *Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.* Proceedings of the ACL-02 conference on Empirical methods in natural language processing 10:1-8. Association for Computational Linguistics.

Aitao Chen. 2013, Private Communication.

Ran El-Yaniv and Yair Wiener. 2010. *On the foundations of noise-free selective classification.* The Journal of Machine Learning Research. 99: 1605-1641. MIT Press.

Jenny Rose Finkel, Christopher D Manning and Andrew Y Ng, 2006. *Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines.* Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp .618—626. Association for Computational Linguistics.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. *Building a Large Annotated Corpus of English: The Penn Treebank.* Computational Linguistics 19(2): 313-330.

Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wentau Yih, Lucy Vanderwende and Colin Cherry. 2012. *MSR SPLAT, a language analysis toolkit.* Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstration Session. Pages 21-24.

Matthew Richardson, Christopher .J.C. Burges and Erin Renshaw. 2013. *MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text.* EMNLP. http://research.microsoft.com/mct

Paul Viola and Michael Jones. 2001. *Robust real-time object detection.* Second international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Mitchell Marcus, Ann Taylor, and others. 2010. *OntoNotes Release 4.0.* Technical Report, BBN Technologies, 2010-12-24.