Published in final edited form as:
Proc Conf Assoc Comput Linguist Meet. 2019 July ; 2019: 2786–2791. doi:10.18653/v1/p19-1267.

We need to talk about standard splits

Kyle Gorman,
City University of New York

Steven Bedrick
Oregon Health & Science University

Abstract

It is standard practice in speech & language technology to rank systems according to performance on a test set held out for evaluation. However, few researchers apply statistical tests to determine whether differences in performance are likely to arise by chance, and few examine the stability of system ranking across multiple training-testing splits. We conduct replication and reproduction experiments with nine part-of-speech taggers published between 2000 and 2018, each of which reports state-of-the-art performance on a widely-used "standard split". We fail to reliably reproduce some rankings using *randomly generated* splits. We suggest that randomly generated splits should be used in system comparison.

1 Introduction

Evaluation with a held-out test set is one of the few methodological practices shared across nearly all areas of speech and language processing. In this study we argue that one common instantiation of this procedure—evaluation with a *standard split*— is insufficient for system comparison, and propose an alternative based on multiple *random splits*.

Standard split evaluation can be formalized as follows. Let G be a set of ground truth data, partitioned into a training set G_{train}, a development set G_{dev} and a test (evaluation) set G_{test}. Let S be a system with arbitrary parameters and hyperparameters, and let $\mathfrak{M}$ be an evaluation metric. Without loss of generality, we assume that $\mathfrak{M}$ is a function with domain $G \times S$ and that higher values of $\mathfrak{M}$ indicate better performance. Furthermore, we assume a supervised training scenario in which the free parameters of S are set so as to maximize $\mathfrak{M}(Gtrain; S)$, optionally tuning hyperparameters so as to maximize $\mathfrak{M}(Gdev; S)$. Then, if S_1 and S_2 are competing systems so trained, we prefer S_1 to S_2 if and only if $\mathfrak{M}(Gtest; S_1) > \mathfrak{M}(Gtest; S_2)$.

1.1 Hypothesis testing for system comparison

One major concern with this procedure is that it treats $\mathfrak{M}(G_{test}, S_1)$ and $\mathfrak{M}(G_{test}, S_2)$ as exact quantities when they are better seen as estimates of random variables corresponding to true system performance. In fact many widely used evaluation metrics, including accuracy and F-score, have known statistical distributions, allowing hypothesis testing to be used for system comparison.

kgorman@gc.cuny.edu, bedricks@ohsu.edu.

For instance, consider the comparison of two systems S_1 and S_2 trained and tuned to maximize accuracy. The difference in test accuracy, $\hat{\delta} = \mathscr{M}(G_{test}, S_1) - \mathscr{M}(G_{test}, S_2)$ can be thought of as estimate of some latent variable δ representing the true difference in system performance. While the distribution of $\hat{\delta}$ is not obvious, the probability that there is no population-level difference in system performance (i.e., $\delta = 0$) can be computed indirectly using McNemar's test (Gillick and Cox, 1989). Let $n_{1>2}$ be the number of samples in G_{test} which S_1 correctly classifies but S_2 misclassifies, and $n_{2>1}$ be the number of samples which S_1 misclassifies but S_2 correctly classifies. When $\delta = 0$, roughly half of the disagreements should favor S_1 and the other half should favor S_2. Thus, under the null hypothesis, $n_{1>2} \sim \text{Bin}(n, .5)$ where $n = n_{1>2} + n_{2>1}$. And, the (one-sided) probability of the null hypothesis is the probability of sampling $n_{1>2}$ from this distribution. Similar methods can be used for other evaluation metrics, or a reference distribution can be estimated with bootstrap resampling (Efron, 1981).

Despite this, few recent studies make use of statistical system comparison. Dror et al. (2018) survey statistical practices in all long papers presented at the 2017 meeting of the Association for Computational Linguistics (ACL), and all articles published in the 2017 volume of the *Transactions of the ACL*. They find that the majority of these works do not use appropriate statistical tests for system comparison, and many others do not report which test(s) were used. We hypothesize that the lack of hypothesis testing for system comparison may lead to type I error, the error of rejecting a true null hypothesis. As it is rarely possible to perform the necessary hypothesis tests from published results, we evaluate this risk using a replication experiment.

1.2 Standard vs. random splits

Furthermore, we hypothesize that standard split methodology may be insufficient for system evaluation. While evaluations based on standard splits are an entrenched practice in many areas of natural language processing, the static nature of standard splits may lead researchers to unconsciously "overfit" to the vagaries of the training and test sets, producing poor generalization. This tendency may also be amplified by *publication bias* in the sense of Scargle (2000). The field has chosen to define "state of the art" performance as "the best performance on a standard split", and few experiments which do not report improvements on a standard split are ultimately published. This effect is likely to be particularly pronounced on highly-saturated tasks for which system performance is near ceiling, as this increases the prior probability of the null hypothesis (i.e., of no difference). We evaluate this risk using a series of reproductions.

1.3 Replication and reproduction

In this study we perform a replication and a series of reproductions. These techniques were until recently quite rare in this field, despite the inherently repeatable nature of most natural language processing experiments. Researchers attempting replications or reproductions have reported problems with availability of data (Mieskes, 2017; Wieling et al., 2018) and software (Pedersen, 2008), and various details of implementation (Fokkens et al., 2013; Reimers and Gurevych, 2017; Schluter and Varab, 2018). While we cannot completely avoid these pitfalls, we select a task—English part-of-speech tagging—for which both data and software are abundantly available. This task has two other important affordances for our purposes. First, it is *face-valid*, both in the sense that the equivalence classes defined by POS tags reflect genuine linguistic insights and that standard evaluation metrics such as token and sentence accuracy directly measure the underlying construct. Secondly, POS tagging is *useful* both in zero-shot settings (e.g., Elkahky et al., 2018; Trask et al., 2015) and as a source of features for many downstream tasks, and in both settings, tagging errors are likely to propagate. We release the underlying software under a permissive license.[1]

2 Materials & Methods

2.1 Data

The Wall St. Journal (WSJ) portion of Penn Treebank-3 (LDC99T42; Marcus et al., 1993) is commonly used to evaluate English part-of-speech taggers. In experiment 1, we also use a portion of OntoNotes 5 (LDC2013T19; Weischedel et al., 2011), a substantial subset of the Penn Treebank WSJ data re-annotated for quality assurance.

2.2 Models

We attempted to choose a set of taggers claiming state-of-the-art performance at time of publication. We first identified candidate taggers using the “State of the Art” page for part-of-speech tagging on the ACL Wiki.[2] We then selected nine taggers for which all needed software and external data was available at time of writing. These taggers are described in more detail below.

2.3 Metrics

Our primarily evaluation metric is token accuracy, the percentage of tokens which are correctly tagged with respect to the gold data. We compute 95% Wilson (1927) score confidence intervals for accuracies, and use the two-sided mid-*p* variant (Fagerland et al., 2013) of McNemar’s test for system comparison. We also report out-of-vocabulary (OOV) accuracy—that is, token accuracy limited to tokens not present in the training data—and sentence accuracy, the percentage of sentences for which there are no tagging errors.

3 Results

Table 1 reports statistics for the standard split. The OntoNotes sample is slightly smaller as it omits sentences on financial news, most of which is highly redundant and idiosyncratic. However, the entire OntoNotes sample was tagged by a single experienced annotator, eliminating any annotator-specific biases in the Penn Treebank (e.g., Ratnaparkhi, 1997, 137f.).

[1] http://github.com/kylebgorman/SOTA-taggers
[2] http://aclweb.org/aclwiki/State_of_the_art

Author Manuscript
Author Manuscript
Author Manuscript

3.1 Models

Three models—SVMTool (Giménez and Màrquez, 2004), MElt (Denis and Sagot, 2009), and Morče/COMPOST (Spoustová et al., 2009)— produced substantial compilation or runtime errors. However, we were able to perform replication with the remaining six models:

- **TnT** (Brants, 2000): a second-order (i.e., trigram) hidden Markov model with a suffix-based heuristic for unknown words, decoded with beam search
- Collins (2002) **tagger**: a linear model, features from Ratnaparkhi (1997), perceptron training with weight averaging, decoded with the Viterbi algorithm[3]
- **LAPOS** (Tsuruoka et al., 2011): a linear model, features from Tsuruoka et al. (2009) plus first-order lookahead, perceptron training with weight averaging, decoded locally
- **Stanford tagger** (Manning, 2011): a log-linear bidirectional cyclic dependency network, features from Toutanova et al. (2003) plus distributional similarity features, optimized with OWL-QN, decoded with the Viterbi algorithm
- **NLP4J** (Choi, 2016): a linear model, dynamically induced features, a hinge loss objective optimized with AdaGrad, decoded locally
- **Flair** (Akbik et al., 2018): a bidirectional long short-term memory (LSTM) conditional random fields (CRF) model, contextual string embedding features, a cross-entropy objective optimized with stochastic gradient descent, decoded globally

3.2 Experiment 1: Replication

In experiment 1, we adopt the standard split established by Collins (2002): sections 00–18 are used for training, sections 19–21 for development, and sections 22–24 for testing, roughly a 80%−10%−10% split. We train and evaluate the six remaining taggers using this standard split. For each tagger, we train on the training set and evaluate on the test set. For taggers which support it, we also perform automated hyperparameter tuning on the development set. Results are shown in Table 2. We obtain exact replications for TnT and LAPOS, and for the remaining four taggers, our results are quite close to previously reported numbers. Token accuracy, OOV accuracy, and sentence accuracy give the same ranking, one consistent with published results. For Penn Treebank, McNemar's test on token accuracy is significant for all pairwise comparisons at $\alpha = .05$; for OntoNotes, one comparison is non-significant: LAPOS vs. Stanford ($p = .1366$).

3.3 Experiment 2: Reproduction

We now repeat these analyses across twenty randomly generated 80%–10%–10% splits. After Dror et al. (2017), we use the Bonferroni procedure to control *familywise error rate*, the probability of falsely rejecting at least one true null hypothesis. This is appropriate insofar as each individual trial (i.e, evaluation on a random split) has a non-trivial statistical dependence on other trials. Table 3 reports the number of random splits, out of twenty,

[3]We use an implementation by Yarmohammadi (2014).

where the McNemar test *p*-value is significant after the correction for familywise error rate. This provides a coarse estimate of how often the second system would be likely to significantly outperform the first system given a random partition of similar size. Most of these pairwise comparisons are stable across random trials. However, for example, Stanford tagger is not a significant improvement over LAPOS for nearly all random trials, and in some random trials—two for Penn Treebank, fourteen for OntoNotes—it is in fact worse. Recall also that the Stanford tagger was also not significantly better than LAPOS for OntoNotes in experiment 1.

Figure 1 shows token accuracies across the two experiments. The last row of the figure gives results for an *oracle ensemble* which correctly predicts the tag just in case any of the six taggers predicts the correct tag.

3.4 Error analysis

From experiment 1, we estimate that the last two decades of POS tagging research has produced a 1.28% absolute reduction in token errors. At the same time, the best tagger is 1.16% below the oracle ensemble. Thus we were interested in disagreements between taggers. We investigate this by treating each of the six taggers as separate coders in a collaborative annotation task. We compute per-sentence inter-annotator agreement using Krippendorff's α (Artstein and Poesio, 2008), then manually inspect sentences with the lowest α values, i.e., with the highest rate of disagreement. By far the most common source of disagreement are "headline"-like sentences such as *Foreign Bonds*. While these sentences are usually quite short, high disagreement is also found for some longer headlines, as in the example sentence in table 4; the effect seems to be due more to capitalization than sentence length. Several taggers lean heavily on capitalization cues to identify proper nouns, and thus capitalized tokens in headline sentences are frequently misclassified as proper nouns and vice versa, as are sentence-initial capitalized nouns in general. Most other sentences with low α have local syntactic ambiguities. For example, the word *lining*, acting as a common noun (NN) in the context …*a silver* ——*for the*…, is mislabeled as a gerund (VBG) by two of six taggers.

4 Discussion

We draw attention to two distinctions between the replication and reproduction experiments. First, we find that a system judged to be significantly better than another on the basis of performance on the standard split, does not in outperform that system on re-annotated data or randomly generated splits, suggesting that it is "overfit to the standard split" and does not represent a genuine improvement in performance. Secondly, as can be seen in figure 1, overall performance is slightly higher on the random splits. We posit this to be an effect of randomization at the sentence-level. For example, in the standard split the word *asbestos* occurs fifteen times in a single training set document, but just once in the test set. Such discrepancies are far less likely to arise in random splits.

Diversity of languages, data, and tasks are all highly desirable goals for natural language processing. However, nothing about this demonstration depends on any particularities of the English language, the WSJ data, or the POS tagging task. English is a somewhat challenging language for POS tagging because of its relatively impoverished inflectional morphology and pervasive noun-verb ambiguity (Elkahky et al., 2018). It would not do to use these six taggers for other languages as they are designed for English text and in some cases depend on English-only external resources for feature generation. However, random split experiments could, for instance, be performed for the subtasks of the CoNLL-2018 shared task on multilingual parsing (Zeman et al., 2018).

We finally note that repeatedly training the Flair tagger in experiment 2 required substantial grid computing resources and may not be feasible for many researchers at the present time.

5 Conclusions

We demonstrate that standard practices in system comparison, and in particular, the use of a single standard split, may result in avoidable Type I error. We suggest that practitioners who wish to firmly establish that a new system is truly state-of the-art augment their evaluations with Bonferroni-corrected random split hypothesis testing.

It is said that statistical praxis is of greatest import in those areas of science least informed by theory. While linguistic theory and statistical learning theory both have much to contribute to part-of-speech tagging, we still lack a theory of the tagging task rich enough to guide hypothesis formation. In the meantime, we must depend on system comparison, backed by statistical best practices and error analysis, to make forward progress on this task.

Acknowledgments

We thank Mitch Marcus for valuable discussion of the Wall St. Journal data.

Steven Bedrick was supported by the National Institute on Deafness and Other Communication Disorders of the National Institutes of Health under award number R01DC015999. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Akbik Alan, Blythe Duncan, and Vollgraf Roland. 2018. Contextual string embedding for sequence labeling. In COLING, pages 1638–1649.

Artstein Ron and Poesio Massimo. 2008. Inter-coder agreement for computational linguistics. Computational Linguistics, 34(4):555–596.

Brants Thorsten. 2000. TnT: a statistical part-of-speech tagger. In ANLC, pages 224–231.

Choi Jinho D.. 2016. Dynamic feature induction: The last gist to the state-of-the-art. In NAACL, pages 271–281.

Collins Michael. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In EMNLP, pages 1–8.

Denis Pascal and Sagot Benoît. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In Pacific Asia Conference on Language, Information and Computation, pages 110–119.

Dror Rotem, Baumer Gili, Bogomolov Marina, and Reichart Roi. 2017. Replicability analysis for natural language processing: testing significance with multiple datasets. Transactions of the Association for Computational Linguistics, 5:471–486.

Dror Rotem, Baumer Gili, Shlomov Segev, and Reichart Roi. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In ACL, pages 1383–1392.

Efron Bradley. 1981. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. Biometrika, 68(3):589–599.

Elkahky Ali, Webster Kellie, Andor Daniel, and Pitler Emily. 2018. A challenge set and methods for noun-verb ambiguity. In EMNLP, pages 2562–2572.

Fagerland Morten W., Lydersen Stian, and Laake Petter. 2013. The McNemar test for binary matched-pairs data: mid-*p* and asymptotic are better than exact conditional. BMC Medical Research Methodology, 13:91–91. [PubMed: 23848987]

Fokkens Antske, Marieke van Erp Marten Postma, Pedersen Ted, Vossen Piek, and Freire Nuno. 2013. Off-spring from reproduction problems: What replication failure teaches us. In ACL, pages 1691–1701.

Gillick Larry and Cox Stephen J.. 1989. Some statistical issues in the comparison of speech recognition algorithms. In ICASSP, pages 23–26.

Giménez Jesús and Màrquez Lluís. 2004. SVMTool: A general POS tagger generator based on support vector machines. In LREC, pages 43–46.

Manning Christopher D.. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In CICLing, pages 171–189.

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

Mieskes Margot. 2017. A quantitative study of data in the NLP community. In Workshop on Ethics in NLP, pages 23–29.

Pedersen Ted. 2008. Empiricism is not a matter of faith. Computational Linguistics, 34(3):465–470.

Ratnaparkhi Adwait. 1997. A maximum entropy model for part-of-speech tagging. In EMNLP, pages 133–142.

Reimers Nils and Gurevych Iryna. 2017. Reporting score distributions makes a difference: performance study of LSTM-networks for sequence tagging. In EMNLP, pages 338–348.

Scargle Jeffrey D.. 2000. Publication bias: the "file-drawer problem" in scientific inference. Journal of Scientific Exploration, 14(1):91–106.

Schluter Natalie and Varab Daniel. 2018. When data permutations are pathological: the case of neural natural language inference. In EMNLP, pages 4935–4939.

Drahomíra Spoustová Jan Hajič, Raab Jan, and Spousta Miroslav. 2009. Semi-supervised training for the averaged perceptron POS tagger. In EACL, pages 763–771.

Toutanova Kristina, Klein Dan, Manning Christopher D., and Singer Yoram. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In NAACL, pages 173–180.

Trask Andrew, Michalak Phil, and Liu John. 2015. sense2vec: A fast and accurate method for word sense disambiguation in neural word embeddings. ArXiv preprint arXiv:1511.06388.

Tsuruoka Yoshimasa, Miyao Yusuke, and Kazama Jun'ichi. 2011. Learning with lookahead: can history-based models rival globally optimized models? In CoNLL, pages 238–246.

Tsuruoka Yoshimasa, Tsujii Jun'ichi, and Ananiadou Sophia. 2009. Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In ICNLP-AFNLP, pages 477–485.

Weischedel Ralph, Hovy Eduard, Marcus Mitchell P., Palmer Martha, Belvin Robert, Pradhan Sameer, …, and Nianwen Xue. 2011. OntoNotes: a large training corpus for enhanced processing. In Olive Joseph, Christianson Caitlin, and McCarthy John, editors, Handbook of natural language processing and machine translation, pages 54–63. Springer, New York.

Wieling Martijn, Rawee Josine, and Gertjan van Noord. 2018. Reproducibility in computational linguistics: are we willing to share? Computational Linguistics, 44(4):641–649.

Wilson Edwin B.. 1927. Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association, 22:209–212.

Yarmohammadi Mahsa. 2014. Discriminative training with perceptron algorithm for POS tagging task. Technical Report CSLU-2014–001, Center for Spoken Language Understanding, Oregon Health & Science University.

Zeman Daniel, Popel Martin, Straka Milan, Jan Hajič Joakim Nivre, Ginter Filip, …, and Li Josie. 2018. CoNLL 2018 shared task: multilingual parsing from raw text to Universal Dependencies.

In Proceedings of the CoNLL 2018 shared task: multilingual parsing from raw text to Universal Dependencies, pages 1–21.

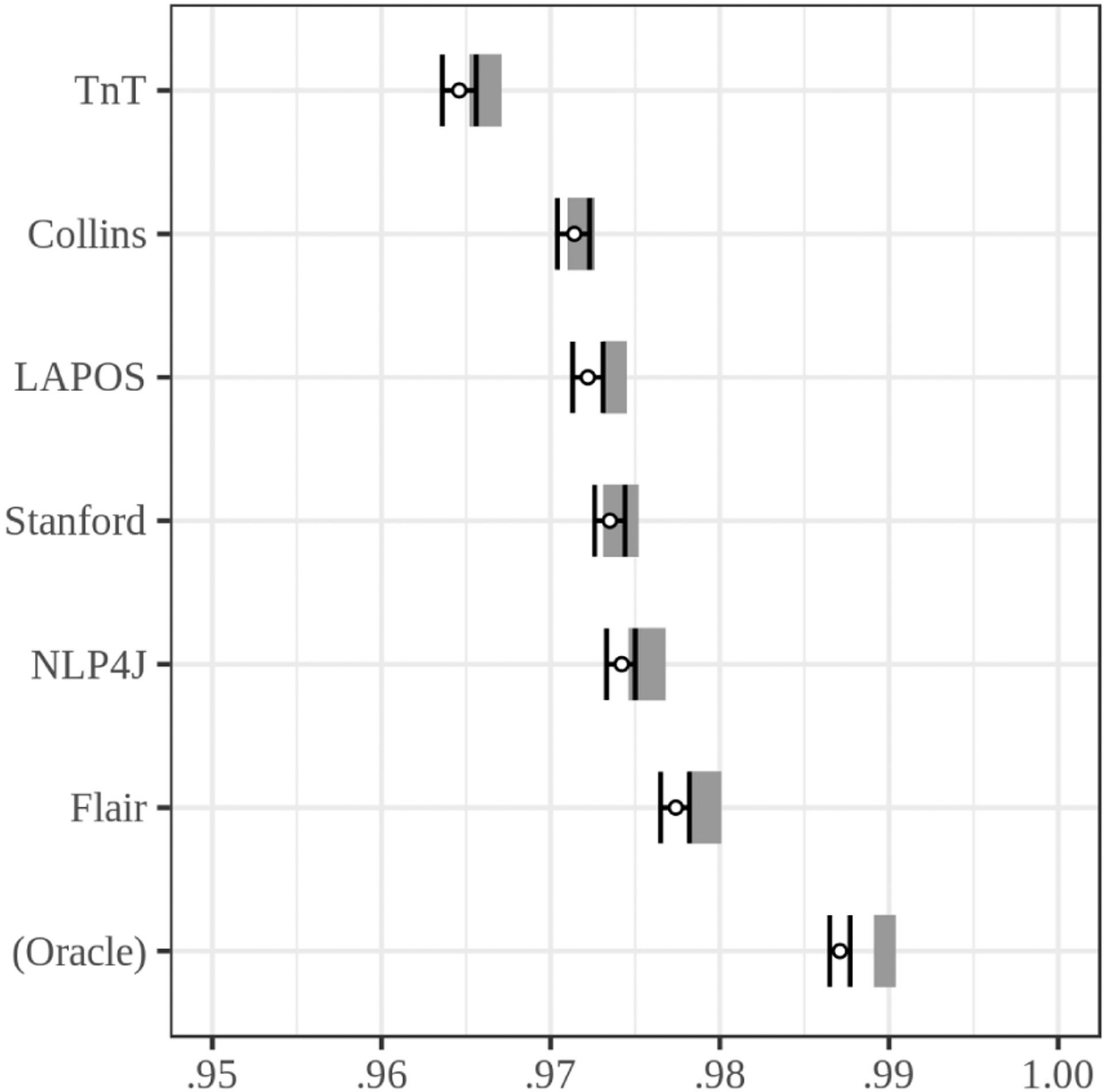

Figure 1:

A visualization of Penn Treebank token accuracies in the two experiments. The whiskers shows accuracy and 95% confidence intervals in experiment 1, and shaded region represents the range of accuracies in experiment 2.

Table 1:

Summary statistics for the standard split.

	# Sentences	# Tokens
	Penn Treebank	
Train.	38,219	912,344
Dev.	5,527	131,768
Test.	5,462	129,654
	OntoNotes	
Train.	28,905	703,955
Dev.	4,051	99,441
Test	4,059	98,277

Table 2:

Previously reported, and replicated, accuracies for the standard split of the WSJ portion of Penn Treebank; we also provide token accuracies for a reproduction with the WSJ portion of OntoNotes.

	Penn Treebank					OntoNotes
	Token			OOV	Sentence	Token
	Reported	Replicated	(95% CIs)	Replicated	Replicated	Reproduced
TnT	.9646	.9646	(.9636, .9656)	.8591	.4771	.9622
Collins	.9711	.9714	(.9704, .9723)	.8789	.5441	.9679
LAPOS	.9722	.9722	(.9713, .9731)	.8874	.5602	.9709
Stanford	.9732	.9735	(.9726, .9744)	.9060	.5710	.9714
NLP4J	.9764	.9742	(.9733, .9750)	.9148	.5756	.9742
Flair	.9785	.9774	(.9765, .9782)	.9287	.6111	.9790

Table 3:

The number of random trials (out of twenty) for which the second system has significantly higher token accuracy than the first after Bonferroni correction. PTB, Penn Treebank; ON, OntoNotes.

	PTB	ON
TnT vs. Collins	20	20
Collins vs. LAPOS	20	7
LAPOS vs. Stanford	1	0
Stanford vs. NLP4J	19	20
NLP4J vs. Flair	20	20

Table 4:

Example error analysis for a Penn Treebank sentence; α = .521.

	Chicken	*Chains*	*Ruffled*	*By*	*Loss*	*of*	*Customers*
Gold	NN	NNS	VBN	IN	NN	IN	NNS
TnT	NNP	NNP	NNP	IN	NN	IN	NNS
Collins	NNP	NNP	NNP	IN	NNP	IN	NNS
LAPOS	NNP	NNP	NNP	NNP	NNP	IN	NNS
Stanford	NNP	NNS	VBN	IN	NN	IN	NNS
NLP4J	NNP	NNPS	NNP	IN	NNP	IN	NNS
Flair	NN	NNS	VBN	IN	NN	IN	NNS