What's in a p-value in NLP?

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy and Hector Martinez
Center for Language Technology
University of Copenhagen
soegaard@hum.ku.dk

Abstract

In NLP, we need to document that our proposed methods perform *significantly* better with respect to standard metrics than previous approaches, typically by reporting p-values obtained by rank- or randomization-based tests. We show that significance results following current research standards are unreliable and, in addition, very sensitive to sample size, covariates such as sentence length, as well as to the existence of multiple metrics. We estimate that under the assumption of perfect metrics and unbiased data, we need a significance cut-off at $\sim$0.0025 to reduce the risk of false positive results to $<5\%$. Since in practice we often have considerable selection bias and poor metrics, this, however, will not do alone.

1 Introduction

In NLP, we try to improve upon state of the art language technologies, guided by experience and intuition, as well as error analysis from previous experiments, and research findings often consist in system comparisons showing that System A is better than System B.

Effect size, i.e., one system's improvements over another, can be seen as a random variable. If the random variable follows a known distribution, e.g., a normal distribution, we can use parametric tests to estimate whether System A is better than System B. If it follows a normal distribution, we can use Student's t-test, for example. Effect sizes in NLP are generally not normally distributed or follow any of the other well-studied distributions (Yeh, 2000; Søgaard, 2013). The standard significance testing methods in NLP are therefore rank- or randomization-based non-parametric tests (Yeh, 2000; Riezler and Maxwell, 2005; Berg-Kirkpatrick et al., 2012). Specifically, most system comparisons across words, sentences or documents use bootstrap tests (Efron and Tibshirani, 1993) or approximate randomization (Noreen, 1989), while studies that compare performance across data sets use rank-based tests such as Wilcoxon's test.

The question we wish to address here is: how likely is a research finding in NLP to be *false*? Naively, we would expect all reported findings to be true, but significance tests have their weaknesses, and sometimes researchers are forced to violate test assumptions and basic statistical methodology, e.g., when there is no *one* established metric, when we can't run our models on full-length sentences, or when data is biased. For example, one such well-known bias from the tagging and parsing literature is what we may refer to as the WSJ FALLACY. This is the false belief that performance on the test section of the Wall Street Journal (WSJ) part of the English Penn treebank is representative for performance on other texts in English. In other words, it is the belief that our samples are always representative. However, (the unawareness of) selection bias is not the only reason research findings in NLP may be false.

In this paper, we critically examine significance results in NLP by simulations, as well as running a series of experiments comparing state-of-the-art POS taggers, dependency parsers, and NER systems, focusing on the sensitivity of p-values to various factors.

Specifically, we address three important factors:

Sample size. When system A is reported to be better than system B, this may not hold across domains (cf. WSJ FALLACY). More importantly, though, it may not even hold on a sub-sample of the test data, or if we added more data points to the test set. Below, we show that in 6/10 of our POS tagger evaluations, significant effects become insignificant by (randomly) adding *more* test data.

Proceedings of the Eighteenth Conference on Computational Language Learning, pages 1–10,
Baltimore, Maryland USA, June 26-27 2014.

Covariates. Sometimes we may bin our results by variables that are actually predictive of the outcome (covariates) (Simmons et al., 2011). In some subfields of NLP, such as machine translation or (unsupervised) syntactic parsing, for example, it is common to report results that only hold for sentences up to some length. If a system A is reported to be better than a system B on sentences up to some length, A need not be better than B, neither for a different length nor in general, since sentence length may actually be predictive of A being better than B.

Multiple metrics. In several subfields of NLP, we have various evaluation metrics. However, if a system A is reported to be better than a system B with respect to some metric M_1, it need not be better with respect to some other metric M_2. We show that even in POS tagging it is sometimes the case that results are significant with respect to one metric, but not with respect to others.

While these caveats should ideally be avoided by reporting significance over varying sample sizes and multiple metrics, some of these effects also stem from the p-value cut-off chosen in the NLP literature. In some fields, p-values are required to be much smaller, e.g., in physics, where the 5 σ criterion is used, and maybe we should also be more conservative in NLP?

We address this question by a simulation of the interaction of type 1 and type 2 error in NLP and arrive at an estimate that more than half of research findings in NLP with $p < 0.05$ are likely to be false, even with a valid metric and in the absence of selection bias. From the same simulations, we propose a new cut-off level at 0.0025 or smaller for cases where the metric can be assumed to be valid, and where there is no selection bias.[1] We briefly discuss what to do in case of selection bias or imperfect metrics.

Note that we do not discuss false discovery rate control or family wise error rate procedures here. While testing with different sample sizes could be be considered multiple hypothesis testing, as pointed out by one of our anonymous reviewers, NLP results should be robust across sample sizes. Note that the $p < 0.0025$ cut-off level corresponds to a Bonferroni correction for a family of $m = 20$ hypotheses.

[1] In many fields, including NLP, it has become good practice to report *actual* p-values, but we still need to understand how significance levels relate to the probability that research findings are false, to interpret such values. The fact that we propose a new cut-off level for the ideal case with perfect metrics and no bias does not mean that we do not recommend reporting actual p-values.

Our contributions

Several authors have discussed significance testing in NLP before us (Yeh, 2000; Riezler and Maxwell, 2005; Berg-Kirkpatrick et al., 2012), but while our discussion touches on many of the same topics, this paper is to the best of our knowledge the first to:

a) show experimentally *how* sensitive p-values are to sample size, i.e., that in standard NLP experiments, significant effects may actually disappear by adding *more* data.

b) show experimentally that multiple metrics and the use of covariates in evaluation increase the probability of positive test results.

c) show that even under the assumption of perfect metrics and unbiased data, as well as our estimates of type 1 and 2 error in NLP, you need at least $p < 0.0025$ to reduce the probability of a research finding being false to be $< 5\%$.

2 Significance testing in NLP

Most NLP metric for comparing system outputs can be shown to be non-normally distributed (Søgaard, 2013) and hence, we generally cannot use statistical tests that rely on such an assumption, e.g., Student's t-test. One alternative to such tests are non-parametric rank-based tests such as Wilcoxon's test. Rank-based tests are sometimes used in NLP, and especially when the number of observations is low, e.g., when evaluating performance across data sets, such tests seem to be the right choice (Demsar, 2006; Søgaard, 2013). The draw-back of rank-based tests is their relatively weak statistical power. When we reduce scores to ranks, we throw away information, and rank-based tests are therefore relatively conservative, potentially leading to high type 2 error rate (β, i.e., the number of false negatives over trials). An alternative, however, are randomization-based tests such as the *bootstrap* test (Efron and Tibshirani, 1993) and *approximate randomization* (Noreen, 1989), which are the *de facto* standards in NLP. In this paper, we follow Berg-Kirkpatrick et al. (2012) in focusing on the bootstrap test. The bootstrap test is non-parametric and stronger than rank-based testing, i.e., introduces fewer type 2 errors. For small samples, however, it does so at the expense of a

higher type 1 error (α, i.e., the number of false positives). The reason for this is that for the bootstrap test to work, the original sample has to capture most of the variation in the population. If the sample is very small, though, this is likely *not* the case. Consequently, with small sample sizes, there is a risk that the calculated p-value will be artificially low—simply because the bootstrap samples are too similar. In our experiments below, we make sure only to use bootstrap when sample size is > 200, unless otherwise stated. In our experiments, we average across 3 runs for POS and NER and 10 runs for dependency parsing.

DOMAIN	#WORDS	TASKS		
		POS	Dep.	NER
CONLL 2007				
Bio	4k	•		
Chem	5k	•		
SWITCHBOARD 4				
Spoken	162k	•		
ENGLISH WEB TREEBANK				
Answers	29k	•	•	
Emails	28k	•	•	
Newsgrs	21k	•	•	
Reviews	28k	•	•	
Weblogs	20k	•	•	
WSJ	40k	•	•	
FOSTER				
Twitter	3k	•		
CONLL 2003				
News	50k			•

Table 1: Evaluation data.

3 Experiments

Throughout the rest of the paper, we use four running examples: a synthetic toy example and three standard experimental NLP tasks, namely POS tagging, dependency parsing and NER. The toy example is supposed to illustrate the logic behind our reasoning and is not specific to NLP. It shows how likely we are to obtain a low p-value for the difference in means when sampling from exactly the same (Gaussian) distributions. For the NLP setups (2-4), we use off-the-shelf models or available runs, as described next.

3.1 Models and data

We use pre-trained models for POS tagging and dependency parsing. For NER, we use the output of the best performing systems from the CoNLL 2003 shared task. In all three NLP setups, we compare the outcome of pairs of systems. The data sets we use for each of the NLP tasks are listed in Table 1 (Nivre et al., 2007a; Foster et al., 2011; Tjong Kim Sang and De Meulder, 2003, LDC99T42; LDC2012T13).

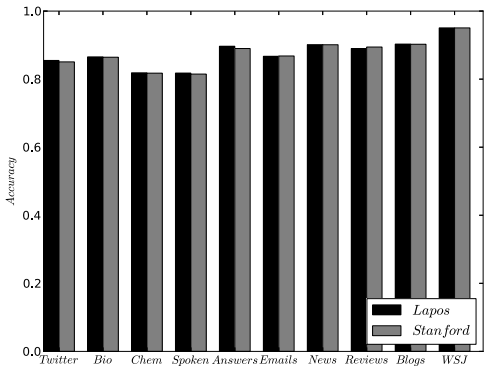

Figure 1: Accuracies of LAPOS VS. STANFORD across 10 data sets.

POS tagging. We compare the performance of two state-of-the-art newswire taggers across 10 evaluation data sets (see Table 1), namely the LAPOS tagger (Tsuruoka et al., 2011) and the STANFORD tagger (Toutanova et al., 2003), both trained on WSJ00–18. We use the publicly available pre-trained models from the associated websites.[2]

Dependency parsing. Here we compare the pre-trained linear SVM MaltParser model for English (Nivre et al., 2007b) to the compositional vector grammar model for the Stanford parser (Socher et al., 2013). For this task, we use the subset of the POS data sets that comes with Stanford-style syntactic dependencies (cf. Table 1), excluding the Twitter data set which we found too small to produce reliable results.

NER. We use the publicly available runs of the two best systems from the CoNLL 2003 shared task, namely FLORIAN (Florian et al., 2003) and CHIEU-NG (Chieu and Ng, 2003).[3]

3.2 Standard comparisons

POS tagging. Figure 1 shows that the LAPOS tagger is marginally better than STANFORD on macro-average, but it is also *significantly* better? If we use the bootstrap test over tagging accuracies, the difference between the two taggers is only significant ($p < 0.05$) in 3/10 cases (see Table 2), namely SPOKEN, ANSWERS and REVIEWS. In two of these cases, LAPOS is significantly better

[2] http://www.logos.ic.i.u-tokyo.ac.jp/~tsuruoka/lapos/ and http://nlp.stanford.edu/software/tagger.shtml

[3] http://www.cnts.ua.ac.be/conll2003/ner/

	TA (b)	UA (b)	SA (b)	SA(w)
Bio	0.3445	0.0430	0.3788	0.9270
Chem	0.3569	0.2566	0.4515	0.9941
Spoken	<0.001	<0.001	<0.001	<0.001
Answers	<0.001	0.0143	<0.001	<0.001
Emails	0.2020	<0.001	0.1622	0.0324
Newsgrs	0.3965	0.0210	0.1238	0.6602
Reviews	0.0020	0.0543	0.0585	0.0562
Weblogs	0.2480	0.0024	0.2435	0.9390
WSJ	0.4497	0.0024	0.2435	0.9390
Twitter	0.4497	0.0924	0.1111	0.7853

Table 2: POS tagging p-values across tagging accuracy (TA), accuracy for unseen words (UA) and sentence-level accuracy (SA) with bootstrap (b) and Wilcoxon (w) ($p < 0.05$ gray-shaded).

	LAS	UAS
Answers	0.020	<0.001
Emails	0.083	<0.001
Newsgroups	0.049	<0.001
Reviews	<0.001	<0.001
Weblogs	<0.001	<0.001
WSJ	<0.001	<0.001

Table 3: Parsing p-values (MALT-LIN VS. STANFORD-RNN) across LAS and UAS ($p < 0.05$ gray-shaded).

than STANFORD, but in one case it is the other way around. If we do a Wilcoxon test over the results on the 10 data sets, following the methodology in Demsar (2006) and Søgaard (2013), the difference, which is ∼0.12% on macro-average, is *not* significant ($p \sim 0.1394$). LAPOS is thus not significantly better than STANFORD across data sets, but as we have already seen, it is significantly better on some data sets. So if we allow ourselves to cherry-pick our data sets and report significance over word-level tagging accuracies, we can at least report significant improvements across a few data sets.

Dependency parsing. Using the bootstrap test over sentences, we get the p-values in Table 3. We see that differences are always significant wrt. UAS, and in most cases wrt. LAS.

NER. Here we use the macro-f_1 as our standard metric. FLORIAN is *not* significantly better than CHIEU-NG with $p < 0.05$ as our cutoff ($p \sim 0.15$). The two systems were also reported to have overlapping confidence intervals in the shared task.

3.3 p-values across metrics

In several NLP subfields, multiple metrics are in use. This happens in dependency parsing where multiple metrics (Schwartz et al., 2011; Tsarfaty et al., 2012) have been proposed in addition to unlabeled and labeled attachment scores, as well as exact matches. Perhaps more famously, in machine translation and summarization it is common practice to use multiple metrics, and there exists a considerable literature on that topic (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005; Clark et al., 2011; Rankel et al., 2011). Even in POS tagging, some report tagging accuracies, tagging accuracies over unseen words, macro-averages over sentence-level accuracies, or number of exact matches.

The existence of several metrics is not in itself a problem, but if researchers can cherry-pick their favorite metric when reporting results, this increases the *a priori* chance of establishing significance. In POS tagging, most papers report significant improvements over tagging accuracy, but some report significant improvements over tagging accuracy of unknown words, e.g., Denis and Sagot (2009) and Umansky-Pesin et al. (2010). This corresponds to the situation in psychology where researchers cherry-pick between several dependent variables (Simmons et al., 2011), which also increases the chance of finding a significant correlation.

Toy example. We draw two times 100 values from identical $(0, 1)$-Gaussians 1000 times and calculate a t-test for two independent samples. This corresponds to testing the effect size between two systems on a 1000 randomly chosen test sets with $N = 100$. Since we are sampling from the same distribution, the chance of $p < \kappa$ should be smaller than κ. In our simulation, the empirical chance of obtaining $p < 0.01$ is .8%, and the chance of obtaining $p < 0.05$ is 4.8%, as expected. If we simulate a free choice between two metrics by introducing choice between a pair of samples and a distorted copy of that pair (inducing random noise at 10%), simulating the scenario where we have a perfect metric and a suboptimal metric, the chance of obtaining $p < 0.05$ is 10.0%. We see a significant correlation ($p < 0.0001$) between Pearson's ρ between the two metrics, and the p-value. The less the two metrics are correlated, the more likely we are to obtain $p < 0.05$. If we allow for a choice between two metrics, the chance of finding a significant difference increases considerably. If the two metrics are identical, but independent (introducing a free choice between two pairs of samples), we have

$P(A \vee B) = P(A) + P(B) - P(A)P(B)$, hence the chance of obtaining $p < 0.01$ is 1.9%, and the chance of obtaining $p < 0.05$ is 9.75%.

POS tagging. In our POS-tagging experiments, we saw a significant improvement in 3/10 cases following the standard evaluation methodology (see Table 2). If we allow for a choice between tagging accuracy and sentence-level accuracy, we see a significant improvement in 4/10 cases, i.e., for 4/10 data sets the effect is significance wrt. at least one metric. If we allow for a free choice between all three metrics (TA, UA, and SA), we observe significance in 9/10 cases. This way the existence of multiple metrics almost guarantees significant differences. Note that there are only two data sets (*Answers* and *Spoken*), where *all* metric differences appear significant.

Dependency parsing. While there are multiple metrics in dependency parsing (Schwartz et al., 2011; Tsarfaty et al., 2012), we focus on the two standard metrics: labeled (LAS) and unlabeled attachment score (UAS) (Buchholz and Marsi, 2006). If we just consider the results in Table 3, i.e., only the comparison of MALT-LIN VS. STANFORD-RNN, we observe significant improvements in all cases, if we allow for a free choice between metrics. Bod (2000) provides a good example of a parsing paper evaluating models using different metrics on different test sets. Chen et al. (2008), similarly, only report UAS.

NER. While macro-f_1 is fairly standard in NER, we do have several available multiple metrics, including the unlabeled f_1 score (collapsing all entity types), as well as the f_1 scores for each of the individual entity types (see Derczynski and Bontcheva (2014) for an example of only reporting f_1 for one entity type). With macro-f_1 and f_1 for the individual entity types, we observe that, while the average p-value for bootstrap tests over five runs is around 0.15, the average p-value with a free choice of metrics is 0.02. Hence, if we allow for a free choice of metrics, FLORIAN comes out significantly better than CHIEU-NG.

3.4 p-values across sample size

We now show that p-values are sensitive to sample size. While it is well-known that studies with low statistical power have a reduced chance of detecting true effects, studies with low statistical power are also more likely to introduce false positives (Button et al., 2013). This, combined with the fact that free choice between different sample sizes also increases the chance of false positives (Simmons et al., 2011), is a potential source of error in NLP.

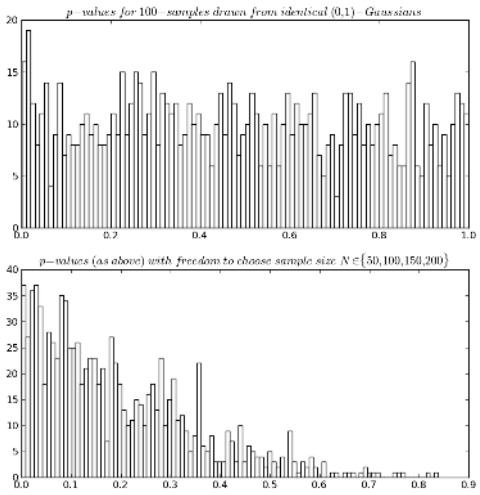

Figure 2: The distribution of p-values with (above) and without (below) multiple metrics.

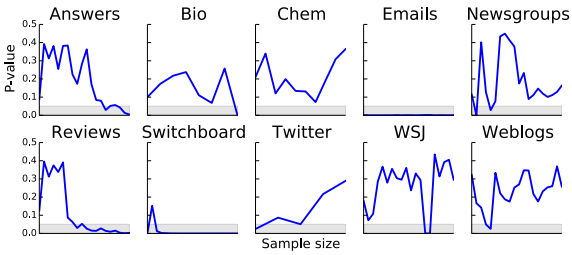

Figure 3: POS tagging p-values varying sample sizes ($p < 0.05$ shaded).

Toy example. The plot in Figure 2 shows the distribution of p-values across 1000 bootstrap tests (above), compared to the distribution of p-values with a free choice of four sample sizes. It is clear that the existence of multiple metrics makes the probability of a positive result much higher.

POS tagging. The same holds for POS tagging. We plot the p-values across various sample sizes in Figure 3. Note that even when we ignore the smallest sample size (500 words), where results may be rather unreliable, it still holds that for *Twitter*, *Answers*, *Newsgrs*, *Reviews*, *Weblogs* and *WSJ*, i.e., more than half of the data sets, a significant result ($p < 0.05$) becomes insignificant by *increasing* the sample size. This shows how unreliable significance results in NLP with cut-off $p < 0.05$ are.

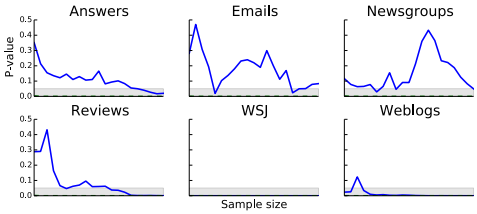

Figure 4: Parsing p-values varying sample sizes ($p < 0.05$ shaded)

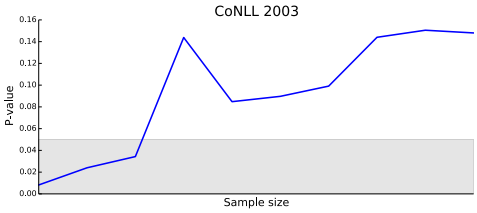

Figure 5: NER p-values varying sample sizes ($p < 0.05$ shaded)

Dependency parsing. We performed similar experiments with dependency parsers, seeing much the same picture. Our plots are presented in Figure 4. We see that while effect sizes are always significant wrt. UAS, LAS differences become significant when adding more data in 4/6 cases. An alternative experiment is to see how often a bootstrap test at a particular sample size comes out significant. The idea is to sample, say, 10% of the test data 100 times and report the ratio of positive results. We only present the results for MALT-LIN VS. STANFORD-RNN in Table 4, but the full set of results (including comparisons of more MaltParser and Stanford parser models) are made available at `http://lowlands.ku.dk`.

For MALT-LIN VS. STANFORD-RNN differences on the full *Emails* data set are consistently insignificant, but on small sample sizes we do get significant test results in more than 1/10 cases. We see the same picture with *Newsgrs* and *Reviews*. On *Weblogs* and *WSJ*, the differences on the full data sets are consistently significant, but here we see that the test is underpowered at small sample sizes. Note that we use bootstrap tests over sentences, so results with small samples may be somewhat unreliable. In sum, these experiments show how small sample sizes not only increase the chance of false negatives, but also the chance of false positives (Button et al., 2013).

NER. Our plots for NER are presented in Figure 5. Here, we see significance at small sample sizes, but the effect disappears with more data. This is an example of how underpowered studies may introduce false positives (Button et al., 2013).

3.5 p-values across covariates

Toy example. If we allow for a choice between two subsamples, using a covariate to single out a subset of the data, the chance of finding a significant difference increases. Even if we let the subset be a random 50-50 split, the chance of obtaining $p < 0.01$ becomes 2.7%, and the chance of obtaining $p < 0.05$ is 9.5%. If we allow for both a choice of dependent variables *and* a random covariate, the chance of obtaining $p < 0.01$ is 3.7%, and the chance of obtaining $p < 0.05$ is 16.2%. So identical Gaussian variables will appear significantly different in 1/6 cases, if our sample size is 100, and if we are allowed a choice between two identical, but independent dependent variables, and a choice between two subsamples provided by a random covariate.

POS We see from Figure 6 that p-values are also very sensitive to sentence length cut-offs. For instance, LAPOS is significantly ($p < 0.05$) better than STANFORD on sentences shorter than 16 words in EMAILS, but not on sentences shorter than 14 words. On the other hand, when longer sentences are included, e.g., up to 22 words, the effect no longer appears significant. On full sentence length, four differences seem significant, but if we allow ourselves to cherry-pick a maximum sentence length, we can observe significant differences in 8/10 cases.

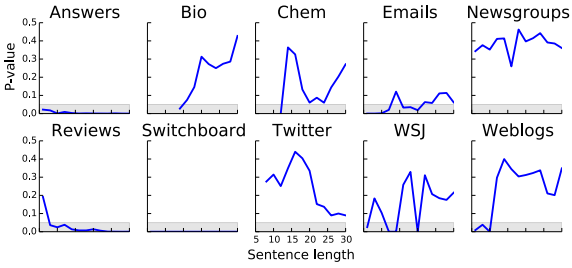

Figure 6: POS tagging p-values varying sentence length ($p < 0.05$ shaded)

We observe similar results in **Dependency parsing** and **NER** when varying sentence length, but do not include them here for space reasons. The results are available at `http://lowlands.ku.dk`. We also found that other covariates are used in evaluations of dependency parsers and NER systems. In dependency parsing, for example, parsers can either be evaluated

N	*Emails*		*Newsgrs*		*Reviews*		*Weblogs*		*WSJ*	
	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
10%	14 %	100 %	9 %	100 %	33%	100 %	42 %	99 %	28 %	75 %
25%	15 %	100 %	23 %	100 %	52%	100 %	68 %	100 %	27 %	98 %
50%	19 %	100 %	25 %	100 %	78%	100 %	100 %	100 %	60 %	100 %
75%	22 %	100 %	41 %	100 %	97%	100 %	100 %	100 %	80 %	100 %
100%	0 %	100 %	36 %	100 %	100%	100 %	100 %	100 %	100 %	100 %

Table 4: Ratio of positive results ($p < 0.05$) for MALT-LIN VS. STANFORD-RNN at sample sizes (N)

.

on naturally occurring text such as in our experiments or at tailored test suites, typically focusing on hard phenomena (Rimell et al., 2009). While such test suites are valuable resources, cf. Manning (2011), they do introduce free choices for researchers, increasing the *a priori* chance of positive results. In NER, it is not uncommon to leave out sentences *without* any entity types from evaluation data. This biases evaluation toward high recall systems, and the choice between including them or not increases chances of positive results.

4 How likely are NLP findings to be false?

The previous sections have demonstrated how many factors can contribute to reporting an erroneously significant result. Given those risks, it is natural to wonder how likely we are as a field to report false positives. This can be quantified by the positive predictive value (PPV), or probability that a research finding is true. PPV is defined as

$$\frac{(1-\beta)R}{R-\beta R+\alpha} \quad (1)$$

The PPV depends on the type 1 and 2 error rates (α and β) and the ratio of true relations over null relations in the field (R) (Ioannidis, 2005).

R. The likelihood that a research finding is true depends on the ratio of true relations over null relations in the field, usually denoted R (Ioannidis, 2005). Out of the systems that researchers in the field would test out (not rejecting them *a priori*), how many of them are better than the current state of the art? The a priori likelihood of a relation being true, i.e., a new system being better than state of the art, is $R/(R+1)$. Note that while the space of reasonably motivated methods may seem big to researchers in the field, there is often more than one method that is better than the current state of the art. Obviously, as the state of the art improves, R drops. On the other hand, if R becomes very low, researchers are likely to move on to new applications where R is higher.

The **type 1 error rate** (α) is also known as the false positive rate, or the likelihood to accept a non-significant result. Since our experiments are fully automated and deterministic, and precision usually high, the type 1 error rate is low in NLP. What is not always appreciated in the field is that this should lead us to expect true effects to be highly significant with very low p-values, much like in physics. The **type 2 error rate** (β) is the false negative rate, i.e., the likelihood that a true relation is never found. This factors into the recall of our experimental set-ups.

So what values should we use to estimate PPV? Our estimate for R (how often reasonable hypotheses lead to improvements over state of the art) is around 0.1. This is based on a sociological rather than an ontological argument. With $\alpha = 0.05$ and $R = 0.1$, researchers get positive results in $R + (1-R)\alpha$ cases, i.e., $\sim 1/7$ cases. If researchers needed to test more than 7 approaches to "hit the nail", they would never get to write papers. With $\alpha = 0.05$, and β set to 0.5, we find that the probability of a research finding being true – given there is *no* selection bias and with perfectly valid metrics – is just 50%:

$$\begin{aligned} PPV &= \frac{(1-\beta)R}{R-\beta R+\alpha} \\ &= \frac{0.5\times 0.1}{0.1-0.05+0.05} = \frac{0.05}{0.1} = 0.5 \end{aligned} \quad (2)$$

In other words, if researchers do a perfect experiment and report $p < 0.05$, the chance of that finding being true is the chance of seeing tail when flipping a coin. With $p < 0.01$, the chance is 5/6, i.e., the chance of not getting a 3 when rolling a die. Of course these parameters are somewhat arbitrary. Figure 7 shows PPV for various values of α.

In the experiments in Section 3, we consistently used the standard p-value cut-off of 0.05. However, our experiments have shown that significance results at this threshold are unreliable and very sensitive to the choice of sample size, covariates, or metrics. Based on the curves in Figure 7, we

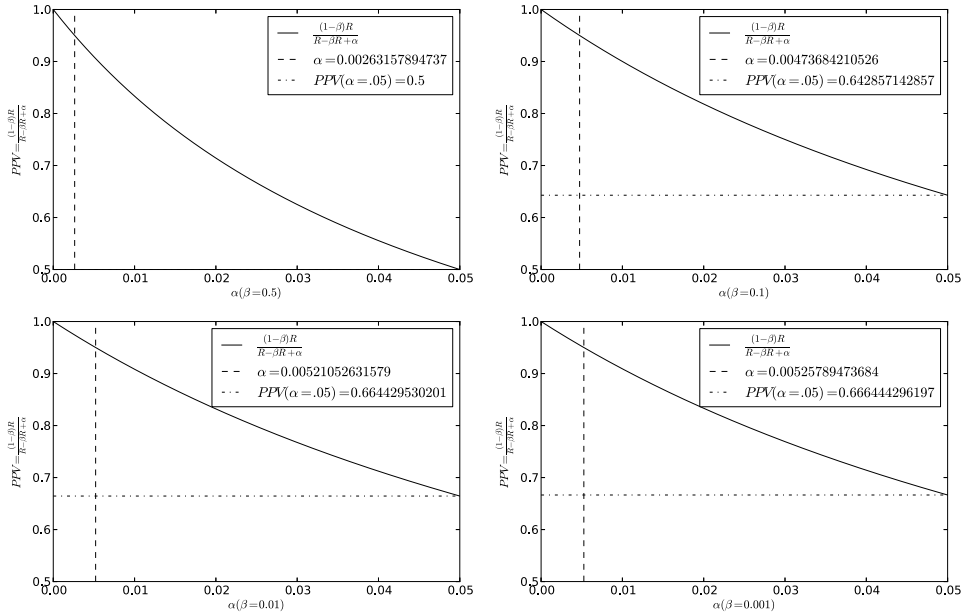

Figure 7: PPV for different α (horizontal line is PPV for $p = 0.05$, vertical line is α for PPV=0.95).

could propose a p-value cut-off at $p < 0.0025$. This is the cut-off that – in the absence of bias and with perfect metrics – gives us the level of confidence we expect as a research community, i.e., $PPV = 0.95$. Significance results would thus be more reliable and reduce type 1 error.

5 Discussion

Incidentally, the $p < 0.0025$ cut-off also leads to a 95% chance of seeing the same effect on held-out test data in Berg-Kirkpatrick et al. (2012) (see their Table 1, first row). The caveat is that this holds only in the absence of bias and with perfect metrics. In reality, though, our data sets are often severely biased (Berg-Kirkpatrick et al., 2012; Søgaard, 2013), and our metrics are far from perfect (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005; Schwartz et al., 2011; Tsarfaty et al., 2012). Here, we discuss how to address these challenges.

Selection bias. The WSJ FALLACY (Section 1) has been widely discussed in the NLP literature (Blitzer et al., 2006; Daume III, 2007; Jiang and Zhai, 2007; Plank and van Noord, 2011). But if our test data is biased, how do we test whether System A performs better than System B in general? Søgaard (2013) suggests to predict significance across data sets. This only assumes that data sets are randomly chosen, e.g., *not* all from newswire corpora. This is also standard practice in the machine learning community (Demsar, 2006).

Poor metrics. For tasks such as POS tagging and dependency parsing, our metrics are suboptimal (Manning, 2011; Schwartz et al., 2011; Tsarfaty et al., 2012). System A and System B may perform equally well as measured by some metric, but contribute very differently to downstream tasks. Elming et al. (2013) show how parsers trained on different annotation schemes lead to very different downstream results. This suggests that being wrong with respect to a gold standard, e.g., choosing NP analysis over a "correct" DP analysis, may in some cases lead to better downstream performance. See the discussion in Manning (2011) for POS tagging. One simple approach to this problem is to report results across available metrics. If System A improves over System B wrt. most metrics, we obtain significance against the odds. POS taggers and dependency parsers should also be evaluated by their impact on downstream performance, but of course downstream tasks may also introduce multiple metrics.

6 Conclusion

In sum, we have shown that significance results with current research standards are unreliable, and we have provided a more adequate p-value cut-off under the assumption of perfect metrics and unbi-

ased data. In the cases where these assumptions cannot be met, we suggest reporting significance results across datasets wrt. all available metrics.

Acknowledgements

We would like to thank the anonymous reviewers, as well as Jakob Elming, Matthias Gondan, and Natalie Schluter for invaluable comments and feedback. This research is funded by the ERC Starting Grant LOWLANDS No. 313695.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *EMNLP*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

Rens Bod. 2000. Parsing with the shortest derivation. In *COLING*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *CoNLL*.

Katherine Button, John Ioannidis, Claire Mokrysz, Brian Nosek, Jonathan Flint, Emma Robinson, and Marcus Munafo. 2013. Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14:365–376.

Wenliang Chen, Youzheng Wu, and Hitoshi Isahara. 2008. Learning Reliable Information for Dependency Parsing Adaptation. In *COLING*.

Hai Leong Chieu and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *CoNLL*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *ACL*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*.

Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *PACLIC*.

Leon Derczynski and Kalina Bontcheva. 2014. Passive-aggressive sequence labeling with discriminative post-editing for recognising person entities in tweets. In *EACL*.

Bradley Efron and Robert Tibshirani. 1993. *An introduction to the bootstrap*. Chapman & Hall, Boca Raton, FL.

Jakob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Hector Martinez Alonso, and Anders Søgaard. 2013. Down-stream effects of tree-to-dependency conversions. In *NAACL*.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *CoNLL*.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Josef Le Roux, Joakim Nivre, Deirde Hogan, and Josef van Genabith. 2011. From news to comments: Resources and benchmarks for parsing the language of Web 2.0. In *IJCNLP*.

John Ioannidis. 2005. Why most published research findings are false. *PLoS Medicine*, 2(8):696–701.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *WAS*.

Chris Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *CICLing*.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007a. The CoNLL 2007 Shared Task on Dependency Parsing. In *EMNLP-CoNLL*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. MaltParser: a language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Eric Noreen. 1989. *Computer intensive methods for testing hypotheses*. Wiley.

Kishore Papineni, Salim Roukus, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, Philadelphia, Pennsylvania.

Barbara Plank and Gertjan van Noord. 2011. Effective measures of domain similarity for parsing. In *ACL*.

Peter Rankel, John Conroy, Eric Slud, and Dianne O'Leary. 2011. Ranking human and machine summarization systems. In *EMNLP*.

Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *EMNLP*.

Roy Schwartz, and Omri Abend, Roi Reichart, and Ari Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL*.

Joseph Simmons, Leif Nelson, and Uri Simonsohn. 2011. False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366.

Richard Socher, John Bauer, Chris Manning, and Andrew Ng. 2013. Parsing with compositional vector grammars. In *ACL*.

Anders Søgaard. 2013. Estimating effect size across datasets. In *NAACL*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *In CoNLL*.

Kristina Toutanova, Dan Klein, Chris Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Cross-framework evaluation for statistical parsing. In *EACL*.

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

Shulamit Umansky-Pesin, Roi Reichart, and Ari Rappoport. 2010. A multi-domain web-based algorithm for POS tagging of unknown words. In *COLING*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *ACL*.