An Empirical Investigation of Statistical Significance in NLP

Taylor Berg-Kirkpatrick **David Burkett** **Dan Klein**
Computer Science Division
University of California at Berkeley
{tberg, dburkett, klein}@cs.berkeley.edu

Abstract

We investigate two aspects of the empirical behavior of paired significance tests for NLP systems. First, when one system appears to outperform another, how does significance level relate in practice to the magnitude of the gain, to the size of the test set, to the similarity of the systems, and so on? Is it true that for each task there is a gain which roughly implies significance? We explore these issues across a range of NLP tasks using both large collections of past systems' outputs and variants of single systems. Next, once significance levels are computed, how well does the standard i.i.d. notion of significance hold up in practical settings where future distributions are neither independent nor identically distributed, such as across domains? We explore this question using a range of test set variations for constituency parsing.

1 Introduction

It is, or at least should be, nearly universal that NLP evaluations include statistical significance tests to validate metric gains. As important as significance testing is, relatively few papers have empirically investigated its practical properties. Those that do focus on single tasks (Koehn, 2004; Zhang et al., 2004) or on the comparison of alternative hypothesis tests (Gillick and Cox, 1989; Yeh, 2000; Bisani and Ney, 2004; Riezler and Maxwell, 2005).

In this paper, we investigate two aspects of the empirical behavior of paired significance tests for NLP systems. For example, all else equal, larger metric gains will tend to be more significant. However, what does this relationship look like and how reliable is it? What should be made of the conventional wisdom that often springs up that a certain metric gain is roughly the point of significance for a given task (e.g. 0.4 F1 in parsing or 0.5 BLEU in machine translation)? We show that, with heavy caveats, there are such thresholds, though we also discuss the hazards in their use. In particular, many other factors contribute to the significance level, and we investigate several of them. For example, what is the effect of the similarity between the two systems? Here, we show that more similar systems tend to achieve significance with smaller metric gains, reflecting the fact that their outputs are more correlated. What about the size of the test set? For example, in designing a shared task it is important to know how large the test set must be in order for significance tests to be sensitive to small gains in the performance metric. Here, we show that test size plays the largest role in determining discrimination ability, but that we get diminishing returns. For example, doubling the test size will not obviate the need for significance testing.

In order for our results to be meaningful, we must have access to the outputs of many of NLP systems. Public competitions, such as the well-known CoNLL shared tasks, provide one natural way to obtain a variety of system outputs on the same test set. However, for most NLP tasks, obtaining outputs from a large variety of systems is not feasible. Thus, in the course of our investigations, we propose a very simple method for automatically generating arbitrary numbers of comparable system outputs and we then validate the trends revealed by our synthetic method against data from public competitions. This methodology itself could be of value in, for example, the design of new shared tasks.

Finally, we consider a related and perhaps even more important question that can only be answered empirically: to what extent is statistical significance on a test corpus predictive of performance on other test corpora, in-domain or otherwise? Focusing on constituency parsing, we investigate the relationship between significance levels and actual performance

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 995–1005, Jeju Island, Korea, 12–14 July 2012. ©2012 Association for Computational Linguistics

on data from outside the test set. We show that when the test set is (artificially) drawn i.i.d. from the same distribution that generates new data, then significance levels are remarkably well-calibrated. However, as the domain of the new data diverges from that of the test set, the predictive ability of significance level drops off dramatically.

2 Statistical Significance Testing in NLP

First, we review notation and standard practice in significance testing to set up our empirical investigation.

2.1 Hypothesis Tests

When comparing a new system A to a baseline system B, we want to know if A is better than B on some large population of data. Imagine that we sample a small test set $x = x_1, \ldots, x_n$ on which A beats B by $\delta(x)$. Hypothesis testing guards against the case where A's victory over B was an unlikely event, due merely to chance. We would therefore like to know how likely it would be that a new, independent test set x' would show a similar victory for A *assuming that A is no better than B on the population as a whole*; this assumption is the null hypothesis, denoted H_0.

Hypothesis testing consists of attempting to estimate this likelihood, written $p(\delta(X) > \delta(x)|H_0)$, where X is a random variable over possible test sets of size n that we could have drawn, and $\delta(x)$ is a constant, the metric gain we actually observed. Traditionally, if $p(\delta(X) > \delta(x)|H_0) < 0.05$, we say that the observed value of $\delta(x)$ is sufficiently unlikely that we should reject H_0 (i.e. accept that A's victory was real and not just a random fluke). We refer to $p(\delta(X) > \delta(x)|H_0)$ as p-value(x).

In most cases p-value(x) is not easily computable and must be approximated. The type of approximation depends on the particular hypothesis testing method. Various methods have been used in the NLP community (Gillick and Cox, 1989; Yeh, 2000; Riezler and Maxwell, 2005). We use the paired bootstrap[1] (Efron and Tibshirani, 1993) because it is one of the most widely used (Och, 2003; Bisani and Ney, 2004; Zhang et al., 2004; Koehn, 2004), and because it can be easily applied to any performance metric, even complex metrics like F1-measure or BLEU (Papineni et al., 2002). Note that we could perform the experiments described in this paper using another method, such as the paired Student's t-test. To the extent that the assumptions of the t-test are met, it is likely that the results would be very similar to those we present here.

```
1. Draw $b$ bootstrap samples $x^{(i)}$ of size $n$ by
   sampling with replacement from $x$.
2. Initialize $s = 0$.
3. For each $x^{(i)}$ increment $s$ if $\delta(x^{(i)}) > 2\delta(x)$.
4. Estimate p-value$(x) \approx \frac{s}{b}$
```

Figure 1: The bootstrap procedure. In all of our experiments we use $b = 10^6$, which is more than sufficient for the bootstrap estimate of p-value(x) to stabilize.

2.2 The Bootstrap

The bootstrap estimates p-value(x) though a combination of simulation and approximation, drawing many simulated test sets $x^{(i)}$ and counting how often A sees an accidental advantage of $\delta(x)$ or greater. How can we get sample test sets $x^{(i)}$? We lack the ability to actually draw new test sets from the underlying population because all we have is our data x. The bootstrap therefore draws each $x^{(i)}$ from x itself, sampling n items from x with replacement; these new test sets are called *bootstrap samples*.

Naively, it might seem like we would then check how often A beats B by more than $\delta(x)$ on $x^{(i)}$. However, there's something seriously wrong with these $x^{(i)}$ as far as the null hypothesis is concerned: the $x^{(i)}$ were sampled from x, and so their average $\delta(x^{(i)})$ won't be zero like the null hypothesis demands; the average will instead be around $\delta(x)$. If we ask how many of these $x^{(i)}$ have A winning by $\delta(x)$, about half of them will. The solution is a recentering of the mean – we want to know how often A does *more than $\delta(x)$ better than expected*. We expect it to beat B by $\delta(x)$. Therefore, we count up how many of the $x^{(i)}$ have A beating B by at least $2\delta(x)$.[2] The pseudocode is shown in Figure 1.

[1] Riezler and Maxwell (2005) argue the benefits of approximate randomization testing, introduced by Noreen (1989). However, this method is ill-suited to the type of hypothesis we are testing. Our null hypothesis does not condition on the test data, and therefore the bootstrap is a better choice.

[2] Note that many authors have used a variant where the event tallied on the $x^{(i)}$ is whether $\delta(x^{(i)}) < 0$, rather than $\delta(x^{(i)}) > 2\delta(x)$. If the mean of $\delta(x^{(i)})$ is $\delta(x)$, and if the distribution of $\delta(x^{(i)})$ is symmetric, then these two versions will be equivalent.

As mentioned, a major benefit of the bootstrap is that any evaluation metric can be used to compute $\delta(x)$.[3] We run the bootstrap using several metrics: F1-measure for constituency parsing, unlabeled dependency accuracy for dependency parsing, alignment error rate (AER) for word alignment, ROUGE score (Lin, 2004) for summarization, and BLEU score for machine translation.[4] We report all metrics as percentages.

3 Experiments

Our first goal is to explore the relationship between metric gain, $\delta(x)$, and statistical significance, p-value(x), for a range of NLP tasks. In order to say anything meaningful, we will need to see both $\delta(x)$ and p-value(x) for many pairs of systems.

3.1 Natural Comparisons

Ideally, for a given task and test set we could obtain outputs from all systems that have been evaluated in published work. For each pair of these systems we could run a comparison and compute both $\delta(x)$ and p-value(x). While obtaining such data is not generally feasible, for several tasks there are public competitions to which systems are submitted by many researchers. Some of these competitions make system outputs publicly available. We obtained system outputs from the TAC 2008 workshop on automatic summarization (Dang and Owczarzak, 2008), the CoNLL 2007 shared task on dependency parsing (Nivre et al., 2007), and the WMT 2010 workshop on machine translation (Callison-Burch et al., 2010).

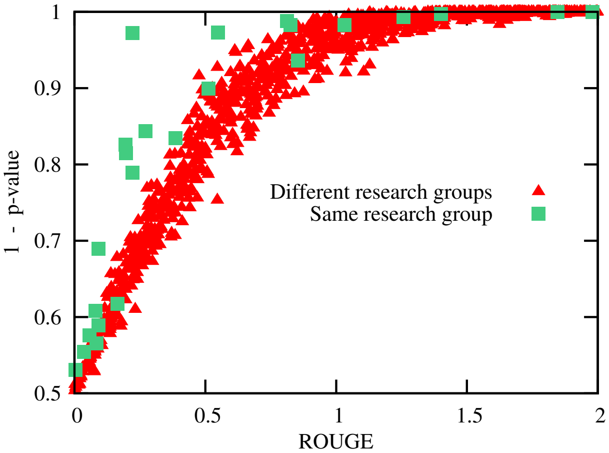

Figure 2: **TAC 2008 Summarization:** Confidence vs. ROUGE improvement on TAC 2008 test set for comparisons between all pairs of the 58 participating systems at TAC 2008. Comparisons between systems entered by the same research group and comparisons between systems entered by different research groups are shown separately.

3.1.1 TAC 2008 Summarization

In our first experiment, we use the outputs of the 58 systems that participated in the TAC 2008 workshop on automatic summarization. For each possible pairing, we compute $\delta(x)$ and p-value(x) on the non-update portion of the TAC 2008 test set (we order each pair so that the gain, $\delta(x)$, is always positive).[5] For this task, test instances correspond to document collections. The test set consists of 48 document collections, each with a human produced summary. Figure 2 plots the ROUGE gain against $1 - $p-value, which we refer to as *confidence*. Each point on the graph corresponds to an individual pair of systems.

As expected, larger gains in ROUGE correspond to higher confidences. The curved shape of the plot is interesting. It suggests that relatively quickly we reach ROUGE gains for which, in practice, significance tests will most likely be positive. We might expect that systems whose outputs are highly correlated will achieve higher confidence at lower metric gains. To test this hypothesis, in Figure 2 we

For cases where the metric linearly decomposes over sentences, the mean of $\delta(x^{(i)})$ is $\delta(x)$. By the central limit theorem, the distribution will be symmetric for large test sets; for small test sets it may not.

[3] Note that the bootstrap procedure given only approximates the true significance level, with multiple sources of approximation error. One is the error introduced from using a finite number of bootstrap samples. Another comes from the assumption that the bootstrap samples reflect the underlying population distribution. A third is the assumption that the mean bootstrap gain is the test gain (which could be further corrected for if the metric is sufficiently ill-behaved).

[4] To save time, we can compute $\delta(x)$ for each bootstrap sample without having to rerun the evaluation metric. For our metrics, sufficient statistics can be recorded for each sentence and then sampled along with the sentences when constructing each $x^{(i)}$ (e.g. size of gold, size of guess, and number correct are sufficient for F1). This makes the bootstrap very fast in practice.

[5] In order to run bootstraps between all pairs of systems quickly, we reuse a random sample counts matrix between bootstrap runs. As a result, we no longer need to perform quadratically many corpus resamplings. The speed-up from this approach is enormous, but one undesirable effect is that the bootstrap estimation noise between different runs is correlated. As a remedy, we set b so large that the correlated noise is not visible in plots.

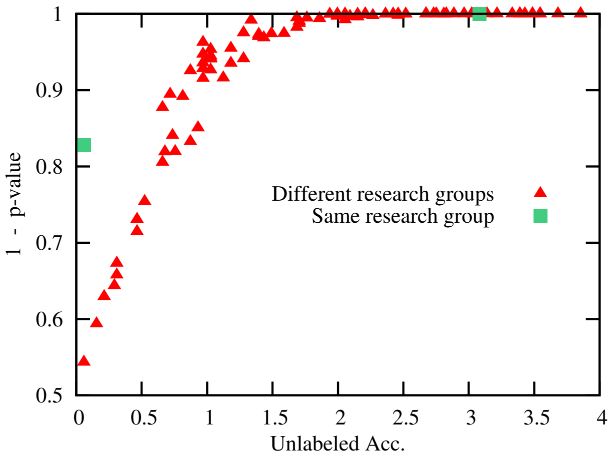

Figure 3: **CoNLL 2007 Dependency parsing:** Confidence vs. unlabeled dependency accuracy improvement on the Chinese CoNLL 2007 test set for comparisons between all pairs of the 21 participating systems in CoNLL 2007 shared task. Comparisons between systems entered by the same research group and comparisons between systems entered by different research groups are shown separately.

separately show the comparisons between systems entered by the same research group and comparisons between systems entered by different research groups, with the expectation that systems entered by the same group are likely to have more correlated outputs. Many of the comparisons between systems submitted by the same group are offset from the main curve. It appears that they do achieve higher confidences at lower metric gains.

Given the huge number of system comparisons in Figure 2, one obvious question to ask is whether we can take the results of all these statistical significance tests and estimate a ROUGE improvement threshold that predicts when future statistical significance tests will probably be significant at the p-value$(x) < 0.05$ level. For example, let's say we take all the comparisons with p-value between 0.04 and 0.06 (47 comparisons in all in this case). Each of these comparisons has an associated metric gain, and by taking, say, the 95th percentile of these metric gains, we get a potentially useful threshold. In this case, the computed threshold is 1.10 ROUGE.

What does this threshold mean? Well, based on the way we computed it, it suggests that if somebody reports a ROUGE increase of around 1.10 *on the exact same test set*, there is a pretty good chance that a statistical significance test would show significance at the p-value$(x) < 0.05$ level. After all, 95% of the borderline significant differences that we've already seen showed an increase of even less than 1.10 ROUGE. If we're evaluating past work, or are in some other setting where system outputs just aren't available, the threshold could guide our interpretation of reports containing only summary scores.

That being said, it is important that we don't overinterpret the meaning of the 1.10 ROUGE threshold. We have already seen that pairs of systems submitted by the same research group and by different research groups follow different trends, and we will soon see more evidence demonstrating the importance of system correlation in determining the relationship between metric gain and confidence. Additionally, in Section 4, we will see that properties of the test corpus have a large effect on the trend. There are many factors are at work, and so, of course, metric gain alone will not fully determine the outcome of a paired significance test.

3.1.2 CoNLL 2007 Dependency Parsing

Next, we run an experiment for dependency parsing. We use the outputs of the 21 systems that participated in the CoNLL 2007 shared task on dependency parsing. In Figure 3, we plot, for all pairs, the gain in unlabeled dependency accuracy against confidence on the CoNLL 2007 Chinese test set, which consists of 690 sentences and parses. We again separate comparisons between systems submitted by the same research group and those submitted by different groups, although for this task there were fewer cases of multiple submission. The results resemble the plot for summarization; we again see a curve-shaped trend, and comparisons between systems from the same group (few that they are) achieve higher confidences at lower metric gains.

3.1.3 WMT 2010 Machine Translation

Our final task for which system outputs are publicly available is machine translation. We run an experiment using the outputs of the 31 systems participating in WMT 2010 on the system combination portion of the German-English WMT 2010 news test set, which consists of 2,034 German sentences and English translations. We again run comparisons for pairs of participating systems. We plot gain in test BLEU score against confidence in Figure 4. In this experiment there is an additional class of compar-

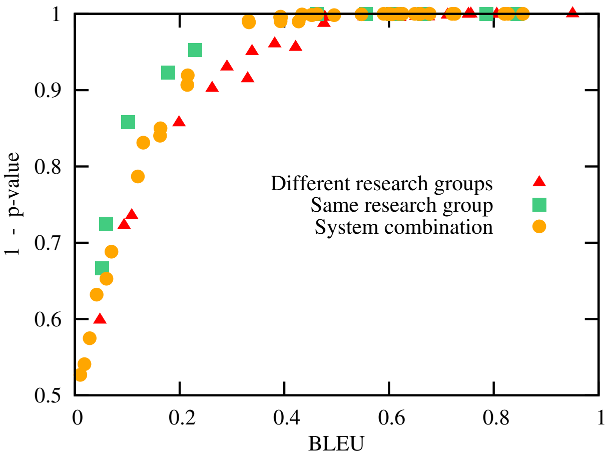

Figure 4: **WMT 2010 Machine translation:** Confidence vs. BLEU improvement on the system combination portion of the German-English WMT 2010 news test set for comparisons between pairs of the 31 participating systems at WMT 2010. Comparisons between systems entered by the same research group, comparisons between systems entered by different research groups, and comparisons between system combination entries are shown separately.

isons that are likely to have specially correlated systems: 13 of the submitted systems are system combinations, and each take into account the same set of proposed translations. We separate comparisons into three sets: comparisons between non-combined systems entered by different research groups, comparisons between non-combined systems entered by the same research group, and comparisons between system-combinations.

We see the same curve-shaped trend we saw for summarization and dependency parsing. Different group comparisons, same group comparisons, and system combination comparisons form distinct curves. This indicates, again, that comparisons between systems that are expected to be specially correlated achieve high confidence at lower metric gain levels.

3.2 Synthetic Comparisons

So far, we have seen a clear empirical effect, but, because of the limited availability of system outputs, we have only considered a few tasks. We now propose a simple method that captures the shape of the effect, and use it to extend our analysis.

3.2.1 Training Set Resampling

Another way of obtaining many different systems' outputs is to obtain implementations of a handful of systems, and then vary some aspect of the training procedure in order to produce many different systems from each implementation. Koehn (2004) uses this sort of amplification; he uses a single machine translation implementation, and then trains it from different source languages. We take a slightly different approach. For each task we pick some fixed training set. Then we generate resampled training sets by sampling sentences with replacement from the original. In this way, we can generate as many new training sets as we like, each of which is similar to the original, but with some variation. For each base implementation, we train a new system on each resampled training set. This results in slightly tweaked trained systems, and is intended to very roughly approximate the variance introduced by incremental system changes during research. We validate this method by comparing plots obtained by the synthetic approach with plots obtained from natural comparisons.

We expect that each new system will be different, but that systems originating from the same base model will be highly correlated. This provides a useful division of comparisons: those between systems built with the same model, and those between systems built with different models. The first class can be used to approximate comparisons of systems that are expected to be specially correlated, and the latter for comparisons of systems that are not.

3.2.2 Dependency Parsing

We use three base models for dependency parsing: MST parser (McDonald et al., 2005), Maltparser (Nivre et al., 2006), and the ensemble parser of Surdeanu and Manning (2010). We use the CoNLL 2007 Chinese training set, which consists of 57K sentences. We resample 5 training sets of 57K sentences, 10 training sets of 28K sentences, and 10 training sets of 14K sentences. Together, this yields a total of 75 system outputs on the CoNLL 2007 Chinese test set, 25 systems for each base model type. The score ranges of all the base models overlap. This ensures that for each pair of model types we will be able to see comparisons where the metric gains are small. The results of the pairwise comparisons of all 75 system outputs are shown in Figure 5, along with the results of the CoNLL 2007 shared task system comparisons from Figure 3.

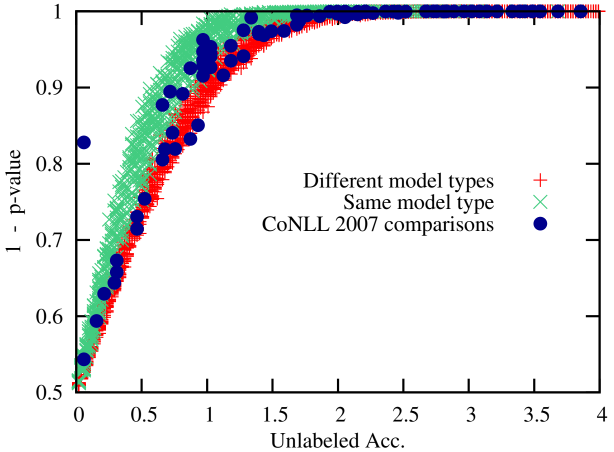

Figure 5: **Dependency parsing:** Confidence vs. unlabeled dependency accuracy improvement on the Chinese CoNLL 2007 test set for comparisons between all pairs of systems generated by using resampled training sets to train either MST parser, Maltparser, or the ensemble parser. Comparisons between systems generated using the same base model type and comparisons between systems generated using different base model types are shown separately. The CoNLL 2007 shared task comparisons from Figure 3 are also shown.

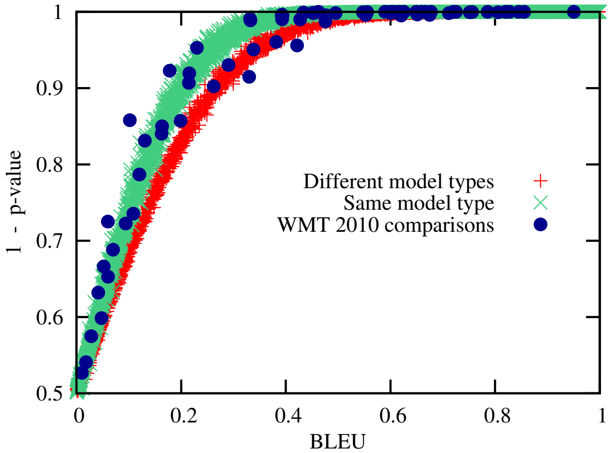

Figure 6: **Machine translation:** Confidence vs. BLEU improvement on the system combination portion of the German-English WMT 2010 news test set for comparisons between all pairs of systems generated by using resampled training sets to train either Moses or Joshua. Comparisons between systems generated using the same base model type and comparisons between systems generated using different base model types are shown separately. The WMT 2010 workshop comparisons from Figure 4 are also shown.

The overlay of the natural comparisons suggests that the synthetic approach reasonably models the relationship between metric gain and confidence. Additionally, the different model type and same model type comparisons exhibit the behavior we would expect, matching the curves corresponding to comparisons between specially correlated systems and standard comparisons respectively.

Since our synthetic approach yields a large number of system outputs, we can use the procedure described in Section 3.1.1 to compute the threshold above which the metric gain is probably significant. For comparisons between systems of the same model type, the threshold is 1.20 unlabeled dependency accuracy. For comparisons between systems of different model types, the threshold is 1.51 unlabeled dependency accuracy. These results indicate that the similarity of the systems being compared is an important factor. As mentioned, rules-of-thumb derived from such thresholds cannot be applied blindly, but, in special cases where two systems are known to be correlated, the former threshold should be preferred over the latter. For example, during development most comparisons are made between incremental variants of the same system. If adding a feature to a supervised parser increases unlabeled accuracy by 1.3, it is useful to be able to quickly estimate that the improvement is probably significant. This still isn't the full story; we will soon see that properties of the test set also play a major role. But first, we carry our analysis to several more tasks.

3.2.3 Machine Translation

Our two base models for machine translation are Moses (Koehn et al., 2007) and Joshua (Li et al., 2009). We use 1.4M sentence pairs from the German-English portion of the WMT-provided Europarl (Koehn, 2005) and news commentary corpora as the original training set. We resample 75 training sets, 20 of 1.4M sentence pairs, 29 of 350K sentence pairs, and 26 of 88K sentence pairs. This yields a total of 150 system outputs on the system combination portion of the German-English WMT 2010 news test set. The results of the pairwise comparisons of all 150 system outputs are shown in Figure 6, along with the results of the WMT 2010 workshop system comparisons from Figure 4.

The natural comparisons from the WMT 2010 workshop align well with the comparisons between synthetically varied models. Again, the different model type and same model type comparisons form distinct curves. For comparisons between systems

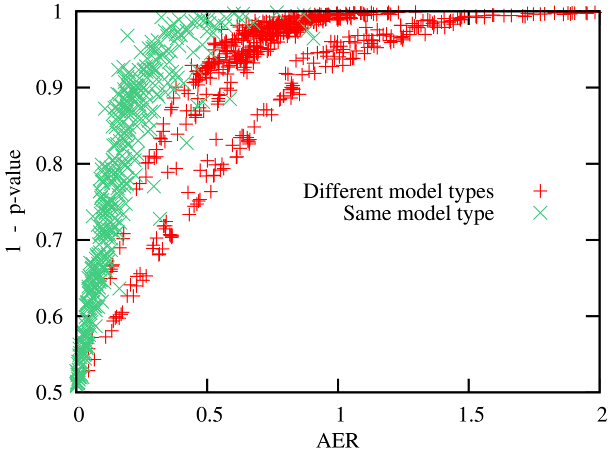

Figure 7: **Word alignment:** Confidence vs. AER improvement on the Hansard test set for comparisons between all pairs of systems generated by using resampled training sets to train either the ITG aligner, the joint HMM aligner, or GIZA++. Comparisons between systems generated using the same base model type and comparisons between systems generated using different base model types are shown separately.

of the same model type the computed p-value < 0.05 threshold is 0.28 BLEU. For comparisons between systems of different model types the threshold is 0.37 BLEU.

3.2.4 Word Alignment

Now that we have validated our simple model of system variation on two tasks, we go on to generate plots for tasks that do not have competitions with publicly available system outputs. The first task is English-French word alignment, where we use three base models: the ITG aligner of Haghighi et al. (2009), the joint HMM aligner of Liang et al. (2006), and GIZA++ (Och and Ney, 2003). The last two aligners are unsupervised, while the first is supervised. We train the unsupervised word aligners using the 1.1M sentence pair Hansard training corpus, resampling 20 training sets of the same size.[6] Following Haghighi et al. (2009), we train the supervised ITG aligner using the first 337 sentence pairs of the hand-aligned Hansard test set; again, we resample 20 training sets of the same size as the original data. We test on the remaining 100 hand-aligned sentence pairs from the Hansard test set.

Unlike previous plots, the points corresponding to comparisons between systems with different base model types form two distinct curves. It turns out that the upper curve consists only of comparisons between ITG and HMM aligners. This is likely due to the fact that the ITG aligner uses posteriors from the HMM aligner for some of its features, so the two models are particularly correlated. Overall, the spread of this plot is larger than previous ones. This may be due to the small size of the test set, or possibly some additional variance introduced by unsupervised training. For comparisons between systems of the same model type the p-value < 0.05 threshold is 0.50 AER. For comparisons between systems of different model types the threshold is 1.12 AER.

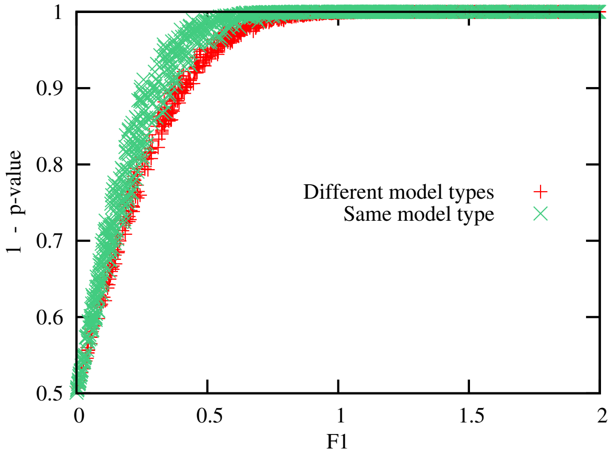

Figure 8: **Constituency parsing:** Confidence vs. F1 improvement on section 23 of the WSJ corpus for comparisons between all pairs of systems generated by using resampled training sets to train either the Berkeley parser, the Stanford parser, or the Collins parser. Comparisons between systems generated using the same base model type and comparisons between systems generated using different base model types are shown separately.

3.2.5 Constituency Parsing

Finally, before we move on to further types of analysis, we run an experiment for the task of constituency parsing. We use three base models: the Berkeley parser (Petrov et al., 2006), the Stanford parser (Klein and Manning, 2003), and Dan Bikel's implementation (Bikel, 2004) of the Collins parser (Collins, 1999). We use sections 2-21 of the WSJ corpus (Marcus et al., 1993), which consists of 38K sentences and parses, as a training set. We resample 10 training sets of size 38K, 10 of size 19K, and 10 of size 9K, and use these to train systems. We test on section 23. The results are shown in Figure 8.

For comparisons between systems of the same

[6] GIZA++ failed to produce reasonable output when trained with some of these training sets, so there are fewer than 20 GIZA++ systems in our comparisons.

model type, the p-value < 0.05 threshold is 0.47 F1. For comparisons between systems of different model types the threshold is 0.57 F1.

4 Properties of the Test Corpus

For five tasks, we have seen a trend relating metric gain and confidence, and we have seen that the level of correlation between the systems being compared affects the location of the curve. Next, we look at how the size and domain of the test set play a role, and, finally, how significance level predicts performance on held out data. In this section, we carry out experiments for both machine translation and constituency parsing, but mainly focus on the latter because of the availability of large test corpora that span more than one domain: the Brown corpus and the held out portions of the WSJ corpus.

4.1 Varying the Size

Figure 9 plots comparisons for machine translation on variously sized initial segments of the WMT 2010 news test set. Similarly, Figure 10 plots comparisons for constituency parsing on initial segments of the Brown corpus. As might be expected, the size of the test corpus has a large effect. For both machine translation and constituency parsing, the larger the corpus size, the lower the threshold for p-value < 0.05 and the smaller the spread of the plot. At one extreme, the entire Brown corpus, which consists of approximately 24K sentences, has a threshold of 0.22 F1, while at the other extreme, the first 100 sentences of the Brown corpus have a threshold of 3.00 F1. Notice that we see diminishing returns as we increase the size of the test set. This phenomenon follows the general shape of the central limit theorem, which predicts that variances of observed metric gains will shrink according to the square root of the test size. Even using the entire Brown corpus as a test set there is a small range where the result of a paired significance test was not completely determined by metric gain.

It is interesting to note that for a fixed test size, the domain has only a small effect on the shape of the curve. Figure 11 plots comparisons for a fixed test size, but with various test corpora.

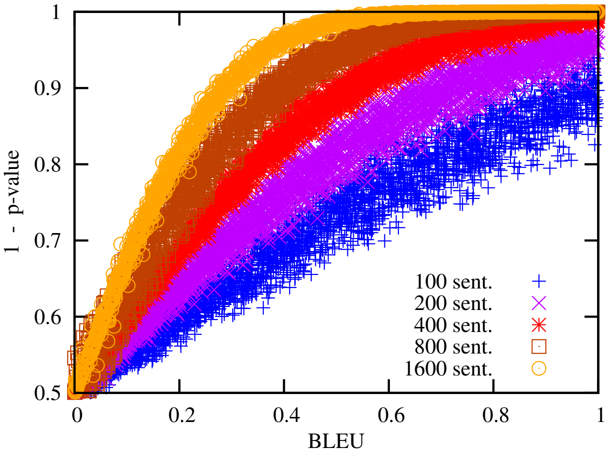

Figure 9: **Machine translation; varying test size:** Confidence vs. BLEU improvement on portions of the German-English WMT 2010 news test set.

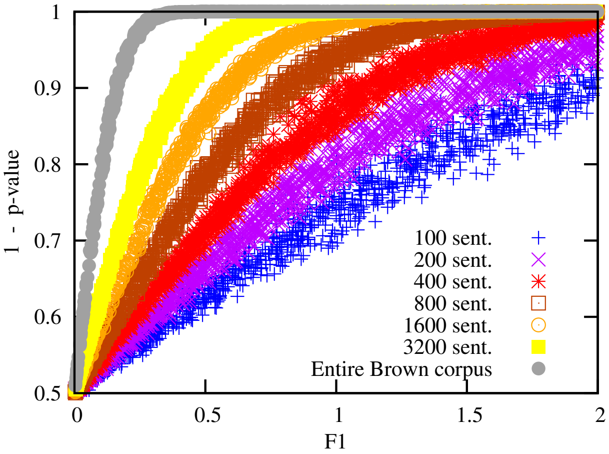

Figure 10: **Constituency parsing; varying test size:** Confidence vs. F1 improvement on portions of the Brown corpus.

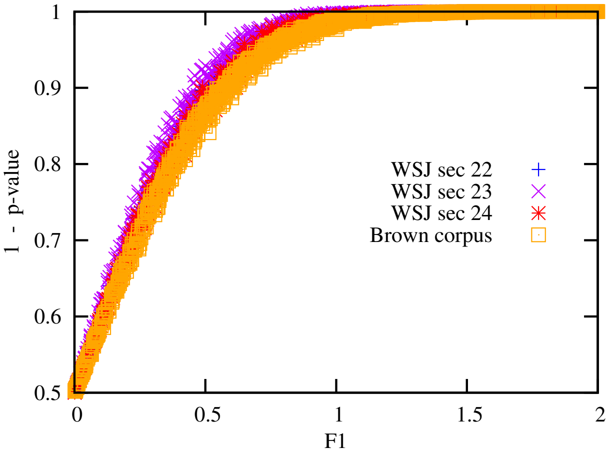

Figure 11: **Constituency parsing; varying domain:** Confidence vs. F1 improvement on the first 1,600 sentences of sections 22, 23, and 24 of the WSJ corpus, and the Brown corpus.

4.2 Empirical Calibration across Domains

Now that we have a way of generating outputs for thousands of pairs of systems, we can check empirically the practical reliability of significance testing. Recall that the bootstrap p-value(x) is an approximation to $p(\delta(X) > \delta(x)|H_0)$. However, we often really want to determine the probability that the new system is better than the baseline on the underlying test distribution or even the distribution from another domain. There is no reason a priori to expect these numbers to coincide.

In our next experiment, we treat the entire Brown corpus, which consists of 24K sentences, as the true population of English sentences. For each system generated in the way described in Section 3.2.5 we compute F1 on *all* of Brown. Since we are treating the Brown corpus as the actual population of English sentences, for each pair of parsers we can say that the sign of the F1 difference indicates which is the truly better system. Now, we repeatedly resample small test sets from Brown, each consisting of 1,600 sentences, drawn by sampling sentences with replacement. For each pair of systems, and for each resampled test set, we compute p-value(x) using the bootstrap. Out of the 4K bootstraps computed in this way, 942 had p-value between 0.04 and 0.06, 869 of which agreed with the sign of the F1 difference we saw on the entire Brown corpus. Thus, 92% of the significance tests with p-value in a tight range around 0.05 correctly identified the better system.

This result is encouraging. It suggests that statistical significance computed using the bootstrap is reasonably well calibrated. However, test sets are almost never drawn i.i.d. from the distribution of instances the system will encounter in practical use. Thus, we also wish to compute how calibration degrades as the domain of the test set changes. In another experiment, we look at how significance near p-value $= 0.05$ on section 23 of the WSJ corpus predicts performance on sections 22 and 24 and the Brown corpus. This time, for each pair of generated systems we run a bootstrap on section 23. Out of all these bootstraps, 58 system pairs had p-value between 0.04 and 0.06. Of these, only 83% had the same sign of F1 difference on section 23 as they did on section 22, 71% the had the same sign on section 23 as on section 24, and 48% the same sign on section 23 as on the Brown corpus. This indicates that reliability degrades as we switch the domain. In the extreme, achieving a p-value near 0.05 on section 23 provides no information about performance on the Brown corpus.

Sec. 23 p-value	% Sys. A > Sys. B		
	Sec. 22	Sec. 24	Brown
0.00125 - 0.0025	97%	95%	73%
0.0025 - 0.005	92%	92%	60%
0.005 - 0.01	92%	85%	56%
0.01 - 0.02	88%	92%	54%
0.02 - 0.04	87%	78%	51%
0.04 - 0.08	83%	74%	48%

Table 1: **Empirical calibration:** p-value on section 23 of the WSJ corpus vs. fraction of comparisons where system A beats system B on section 22, section 24, and the Brown corpus. Note that system pairs are ordered so that A always outperforms B on section 23.

If we intend to use our system on out-of-domain data, these results are somewhat discouraging. How low does p-value(x) have to get before we start getting good information about out-of-domain performance? We try to answer this question for this particular parsing task by running the same domain calibration experiment for several different ranges of p-value. The results are shown in Table 1. From these results, it appears that for constituency parsing, when testing on section 23, a p-value level below 0.00125 is required to reasonably predict performance on the Brown corpus.

It should be considered a good practice to include statistical significance testing results with empirical evaluations. The bootstrap in particular is easy to run and makes relatively few assumptions about the task or evaluation metric. However, we have demonstrated some limitations of statistical significance testing for NLP. In particular, while statistical significance is usually a minimum necessary condition to demonstrate that a performance difference is real, it's also important to consider the relationship between test set performance and the actual goals of the systems being tested, especially if the system will eventually be used on data from a different domain than the test set used for evaluation.

5 Conclusion

We have demonstrated trends relating several important factors to significance level, which include both properties of the systems being compared and properties of the test corpus, and have presented a simple approach to approximating the response of these factors for tasks where large numbers of system outputs are not available. Our results reveal that the relationship between metric gain and statistical significance is complex, and therefore simple thresholds are not a replacement for significance tests. Indeed, we strongly advocate the use of statistical significance testing to validate metric gains in NLP, but also note that informal rules-of-thumb do arise in popular discussion and that, for some settings when previous systems are unavailable, these empirical results can supplement less sensitive unpaired tests (e.g. bar-overlaps-point test) in evaluation of progress. Finally, even formal testing has its limits. We provide cautionary evidence to this effect, showing that the information provided by a test quickly degrades as the target corpus shifts domain.

Acknowledgements

This work was partially supported by NSF fellowships to the first and second authors and by the NSF under grant 0643742.

References

D.M. Bikel. 2004. Intricacies of collins' parsing model. *Computational Linguistics*.

M. Bisani and H. Ney. 2004. Bootstrap estimates for confidence intervals in asr performance evaluation. In *Proc. of ICASSP*.

C. Callison-Burch, P. Koehn, C. Monz, K. Peterson, M. Przybocki, and O.F. Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proc. of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.

M. Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

H.T. Dang and K. Owczarzak. 2008. Overview of the tac 2008 update summarization task. In *Proc. of Text Analysis Conference*.

B. Efron and R. Tibshirani. 1993. *An introduction to the bootstrap*. Chapman & Hall/CRC.

L. Gillick and S.J. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. of ICASSP*.

A. Haghighi, J. Blitzer, J. DeNero, and D. Klein. 2009. Better word alignments with supervised ITG models. In *Proc. of ACL*.

D. Klein and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*.

Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W.N.G. Thornton, J. Weese, and O.F. Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proc. of the Fourth Workshop on Statistical Machine Translation*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Proc. of NAACL*.

C.Y. Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. of the Workshop on Text Summarization*.

M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of EMNLP*.

J. Nivre, J. Hall, and J. Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proc. of LREC*.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*.

E.W. Noreen. 1989. *Computer Intensive Methods for Hypothesis Testing: An Introduction*. Wiley, New York.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL*.

S. Riezler and J.T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

M. Surdeanu and C.D. Manning. 2010. Ensemble models for dependency parsing: cheap and good? In *Proc. of NAACL*.

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

Y. Zhang, S. Vogel, and A. Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system. In *Proc. of LREC*.