A Two-Stage Method for Active Learning of Statistical Grammars

Markus Becker
School of Informatics
University of Edinburgh
m.becker@ed.ac.uk

Miles Osborne
School of Informatics
University of Edinburgh
miles@inf.ed.ac.uk

Abstract

Active learning reduces the amount of manually annotated sentences necessary when training state-of-the-art statistical parsers. One popular method, *uncertainty sampling*, selects sentences for which the parser exhibits low certainty. However, this method does not quantify confidence about the current statistical model itself. In particular, we should be less confident about selection decisions based on low frequency events. We present a novel two-stage method which first targets sentences which cannot be reliably selected using uncertainty sampling, and then applies standard uncertainty sampling to the remaining sentences. An evaluation shows that this method performs better than pure uncertainty sampling, and better than an ensemble method based on bagged ensemble members only.

1 Introduction

State-of-the-art parsers [Collins, 1997; Charniak, 2000] require large amounts of manually annotated training material, such as the Penn Treebank [Marcus *et al.*, 1993], to achieve high performance levels. However, creating such labelled data sets is costly and time-consuming. Active learning promises to reduce this cost by requesting only highly informative examples for human annotation. Methods have been proposed that estimate example informativity by the degree of uncertainty of a single learner as to the correct label of an example [Lewis and Gale, 1994] or by the disagreement of a committee of learners [Seung *et al.*, 1992]. This paper is concerned with reducing the manual annotation effort necessary to train a state-of-the-art lexicalised parser [Collins, 1997].

Uncertainty-based sampling has been successfully applied to the same problem problem [Hwa, 2000]. Here, sentences are selected for manual annotation when the entropy over the probability distribution of competing analyses is high. Entropy quantifies the degree of uncertainty as to the correct analysis of a sentence.

A problem with active learning methods such as uncertainty sampling is that they have no method for dealing with the consequences of low counts. For example, the parse tree probability of the most likely reading in a peaked distribution may depend on a probability which has been unreliably estimated from an as yet rarely observed event. In this case, the model would indicate certainty about a particular analysis where indeed it is not confident. In general, we would like to place less confidence in selection decisions based on entropy over probability distributions involving low frequency events. However, sentences whose predicted parse was selected on the basis of infrequent events may well be informative. Since entropy will not in itself always allow us to reliably select such examples for labelling, we need to consider other mechanisms.

We propose a novel two-stage method which first selects unparsable sentences according to a bagged parser, and applies uncertainty sampling to the remaining sentences using a fully trained parser. Evaluation shows that this method performs better than single parser uncertainty sampling, and better than an ensemble method with bagged ensemble members.

To explain our results, we show empirically that entropy and f-measure are negatively correlated. Thus, selection according to entropy tends to acquire annotations of sentences with low f-measure under the current model. An oracle-based experiment demonstrates that preferably selecting low f-measure sentences is indeed beneficial and explains why uncertainty sampling is successful in general. Furthermore, we find that exactly those sentences which our proposed methods targets show no such correlation between entropy and f-measure. In other words, entropy will not reliably identify informative examples from this subset, even though these sentences have below average f-measure and should be particularly useful. These findings help to explain why the proposed method is a successful strategy.

2 Active Learning Methods

Popular methods for active learning estimate example informativity with the uncertainty of a single classifier or the disagreement of an ensemble of classifiers.

Uncertainty-based sampling (or *tree entropy*) chooses examples with high entropy of the probability distribution for a single parser [Hwa, 2000]:

$$f_M^{te}(s,\tau) = -\sum_{t\in\tau} P_M(t|s)\log P_M(t|s) \qquad (1)$$

where τ is the set of parse trees assigned to sentence s by a stochastic parser with parameter model M. Less spiked distributions have a higher entropy and indicate uncertainty of the parse model as to the correct analysis. Thus, it will be useful to know their true parse tree.

Ensemble-based methods for active learning select examples for which an ensemble of classifiers shows a high degree of disagreement. *Kullback-Leibler divergence to the mean* quantifies ensemble disagreement [Pereira *et al.*, 1993; McCallum and Nigam, 1998]. It is the average Kullback-Leibler divergence between each distribution and the mean of all distributions:

$$f_{\mathcal{M}}^{kl}(s,\tau) = \frac{1}{k}\sum_{M\in\mathcal{M}} D(P_M||P_{avg}) \qquad (2)$$

where $\mathcal{M}$ denotes the set of k ensemble models, P_{avg} is the mean distribution over ensemble members in $\mathcal{M}$, $P_{avg} = \sum_M P_M(t|s)/k$, and $D(\cdot||\cdot)$ is KL-divergence, an information-theoretic measure of the difference between two distributions. It will be useful to acquire the manual annotation of sentences with a high *KL-divergence to the mean*. This metric has been applied for active learning in the context of text classification [McCallum and Nigam, 1998].

3 A Novel Two-Stage Selection Method

Acquiring the correct analysis of a sentence of which the predicted analysis was selected on the basis of infrequent events may well be informative. Since entropy itself will not allow us to reliably select such examples for labelling, we need to consider other mechanisms. A simple, but effective method is to eliminate some infrequent events from the parsing model. Simply bagging the current training set, and retraining the parser on this set allows to identify such examples for labelling.

Bagging is a general machine learning technique that reduces variance of the underlying training methods [Breiman, 1996]. It aggregates estimates from classifiers trained on bootstrap replicates (bags) of the original training data. Creating a bootstrap replicate entails sampling with replacement n examples from a training set of n examples. A bootstrap replicate will not only perturb all event counts to some degree, but will inevitably eliminate some of the low frequency event types.

The proposed method operates in two stages. We first select sentences which are unparsable according to a single bagged version of the parser, but (possibly) parsable under the current fully trained model. From the remaining sentences, we select those with the highest entropy as determined by the fully trained model. We can express this formally as follows:

$$f_{M,M'}^{two}(s,\tau) = \max(f_M^{te}(s,\tau_M), failure(s,M')) \qquad (3)$$

where f_M^{te} is tree entropy according to a fully trained model M, defined in (1). The function $failure(s,M')$ returns infinity when sentence s is parsable given bagged parser model M', and 0 otherwise.

4 Experimental Setup

For our experiment, we employ a state-of-the-art lexicalised parser [Collins, 1997].[1] We employ default settings without expending any effort to optimise parameters towards the considerably smaller training sets involved in active learning.

In common with almost all active learning research, we compare the efficacy of different selection methods by performing simulation experiments. We label sentences of sections 02 - 22 from the Penn WSJ treebank [Marcus *et al.*, 1993], ignoring sentences longer than 40 words.

We report the average over a 5-fold cross-validation to ensure statistical significance of the results. In a single fold, we randomly sample (without replacement) an initial labelled training set of a fixed size – 500 or 2,000 sentences, depending on the experiment – and a test set of 1,000 sentences. The remaining sentences constitute the global pool of unlabelled sentences (ca. 37,000 sentences). For a realistic experiment, we tag the test set with the TnT part-of-speech tagger as input for the parser [Brants, 2000]. We train TnT on 30,000 sentences in the global resource. In a 5-fold cross-validation, the parser has 88.8% labelled precision and 88.6% labelled recall, when trained on 37k sentences and applied to test sets of 1,000 sentences.[2]

We randomly sample (without replacement) a subset of 1,000 sentences from the global pool in each iteration. From this subset, 100 sentences are selected for manual annotation according to the current sample selection method. Then, annotated sentences are added to the training set.

For consistent comparison across methods, we evaluate test set performance of a single parser trained on the entirety of the labeled training data at each step, regardless of the selection method being a single or an ensemble method.

Length balanced sampling For situations such as active learning for parsing, the sentences in question need a variable number of labelling decisions. This may confound sample selection metrics and it is therefore necessary to normalise. For example, tree entropy may be directly normalised by sentence length [Hwa, 2000], or by the binary logarithm of the number of parser readings [Hwa, 2001].

We use the following method to control for sentence length in sample selection: Given a batch size b, we randomly sample b sentences from the pool and record the number e_l of selected examples for sentence length l. Then, for all lengths $l = 1, 2, \ldots 40$, we select from all sentences in the pool of length l the e_l examples with the highest score according to our sample selection metric. Of course, the union of these sets will have b examples again. Since we randomly sampled the batch from the pool, we may assume that the batch distribution reflects the pool distribution, in particular wrt. the distribution of sentence lengths.

[1] We use the flexible reimplementation of the parser by Dan Bikel, developed at the University of Pennsylvania [Bikel, 2004]. It can be obtained at http://www.cis.upenn.edu/~dbikel/

[2] It would be desirable for methodological reasons to automatically tag the global resource, too. However, our corpus split scheme does not leave enough disjoint training material for the tagger, so we use the gold standard tags for the pool sentences.

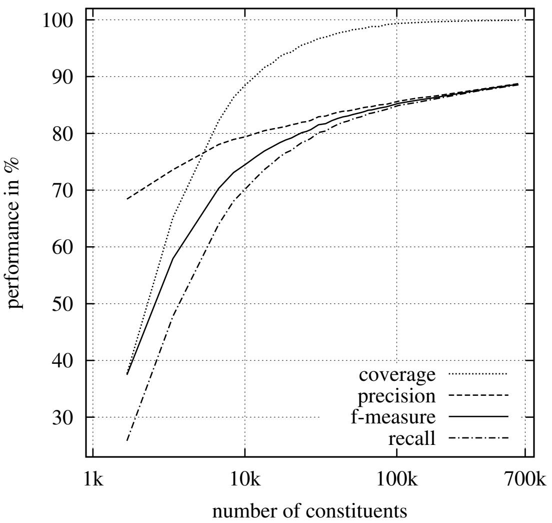

Figure 1: A random sampling learning curve. The maximal training set has 37k sentences (630k constituents). The number of constituents is given on a log scale.

This method effectively reproduces the sentence length profile of the original corpus by construction and therefore guards against the selection of sentence length biased subsets. Furthermore, it is equally applicable for all metrics and allows a direct comparison between metrics. Note that this method is applicable in general for the sample selection of sequential data where one may expect to find correlation between sample length and score.

Relevant evaluation measures Active learning for parsing is typically evaluated in terms of achieving a given f-measure for some amount of labelling expenditure. The cost of acquiring manually annotated training material is given in terms of the number of constituents. F-measure itself is a composite term, being composed of precision and recall [Black *et al.*, 1991]. Fig. 1 shows a learning curve for a random sampling experiment. We see that precision and recall do not increase at the same rate. For this reason, it may well be advantageous to aggressively increase recall with minimal impact on precision (formulate stronger: still want to increase precision). One way of achieving this is to pursue sentences which cannot be parsed.

5 Results

The experiments in this section address the following questions. Is it generally useful to select unparsable sentences for manual annotation? What is the gain of using the novel two-stage method over a state-of-the-art uncertainty-based sample selection method? Given that the two-stage method has a bagged component, how does it compare against a state-of-the-art ensemble-based method which employs bagged ensemble members?

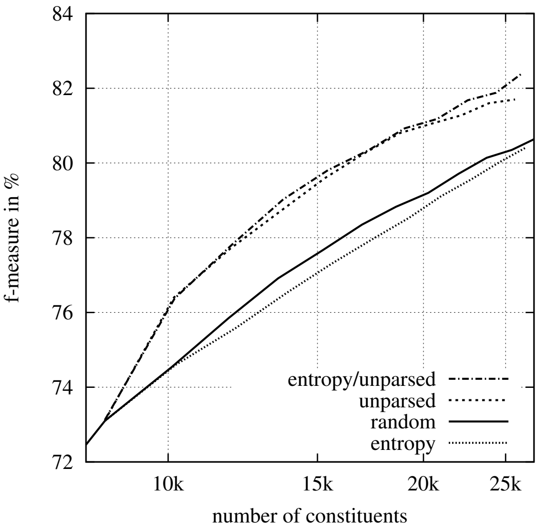

Figure 2: Comparison of f-measure learning curves. Number of constituents is given on a log scale.

Method	Cost	Reduction
random	23200	N/A
entropy/unparsed	16100	30.6%
unparsed	16400	29.4%
entropy	24500	-5.7%

Table 1: Annotation cost to reach 80% f-measure, and reduction over random sampling.

Including/excluding unparsed sentences In this experiment, we compare methods which do or do not include unparsed sentences in the batch of selected examples. Acquiring the correct parse tree of an unparsable sentence increases the size of the model structure of the grammar and, presumably, helps to increase coverage in the test set.

A very simple method, **unparsed**, preferably includes unparsed pool sentences in the batch. Should the number of unparsed sentences fall short of the batch size, we randomly sample parsable sentences from the pool to fill the batch. By contrast, **entropy** only selects parsable sentences with high entropy. The method **entropy/unparsed** preferably selects unparsed pool sentences, and fills the batch with high entropy examples. We may view this method as being composed of a binary parsability component, and a gradual uncertainty component. The baseline is **random**, a parser trained on randomly sampled training sets of different sizes.

We start with an initial training set of 500 randomly sampled sentences, containing 8,400 labeled constituents and continue for 10 rounds until 1,500 sentences have been sampled (ca. 26k constituents). Here, as in all subsequent experiments we employ length balanced sampling, cf. Sec. 4.

Methods **unparsed** and **entropy/unparsed** perform consistently better than **random** (Fig. 2). Note that their performance is nearly identical until more than 20k constituents have been labelled. Method **entropy** performs consistently

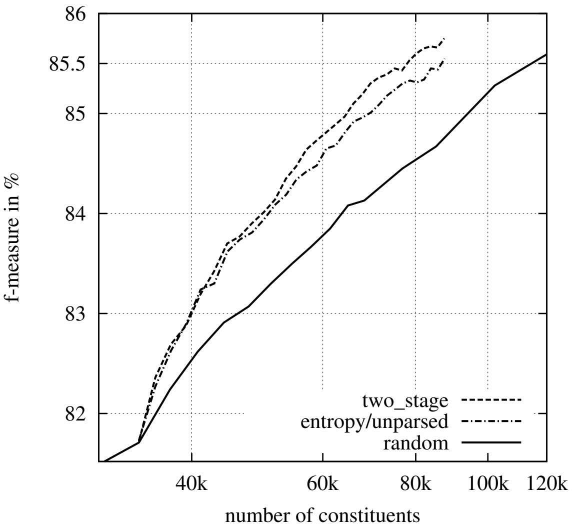

Figure 3: The new two-stage method versus state-of-the-art uncertainty sampling.

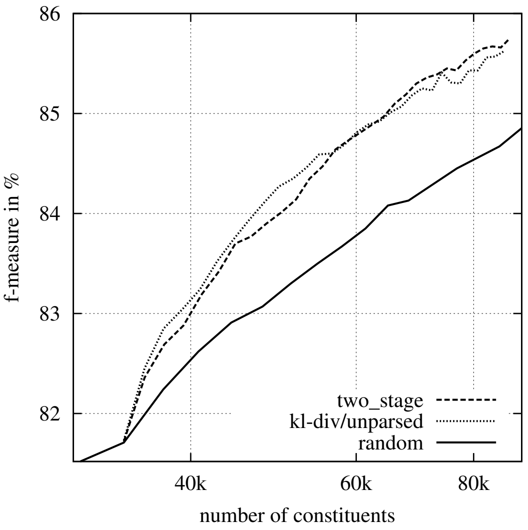

Figure 4: The new two-stage method versus a state-of-the-art ensemble-based method.

Method	Cov	Prec	Rec	Fm
random	95.8	82.0	78.6	80.3
entropy/unparsed	98.3	82.9	81.2	82.1
unparsed	98.5	82.4	80.9	81.7
entropy	94.4	82.8	77.6	80.1

Table 2: Parseval values for different metrics after 25,000 constituents have been annotated.

Method	Cost	Reduction
random	115600	N/A
two_stage	77900	32.6%
entropy/unparsed	86700	25.0%

Table 3: Annotation cost to reach 85.5% f-measure, and reduction over random sampling.

worse than **random**.

Methods **unparsed** and **entropy/unparsed** reduce the amount of labeled constituents necessary to achieve 80% f-measure by around 30% as compared to **random**, while **entropy** actually increases the cost by 5.7% (Tab. 1).

We also compare performance across methods for the same amount of annotation effort. Tab. 2 shows precision, recall, and f-measure after labelling 25k constituents. Methods **unparsed** and **entropy/unparsed** have considerably higher coverage than **random**, **entropy** has lower coverage. While all methods show comparable values for precision, they differ decidedly in their recall values. The two methods which aggressively pursue unparsed sentences, **unparsed** and **entropy/unparsed**, have more than 3% points higher recall than **entropy**, and accordingly higher f-measures than **random** and **entropy**.

These results confirm the importance of including unparsed sentences. Doing so helps achieving better coverage and a higher recall value which directly translates into higher f-measure. Accordingly, all of the following experiments will include unparsed sentences in the batch. The negative result for the purely entropy-based method shows clearly that a naive application of uncertainty sampling may have adverse consequences. It is extremely important to consider which phenomena a selection method is targeting.

A two-stage selection method The novel method addresses problems with sentences which cannot be reliably selected with popular active learning methods. Therefore, we expect a gain in performance. Method **two_stage** preferably includes sentences which are unparsable according to a parser trained on a bagged version of the current training set. Then, the batch is filled with (parsable) sentences which have high entropy according to a second, fully trained parser.

Given that composite methods which preferably select unparsed sentences perform nearly uniformly well, we will now use a considerably larger initial training set in order to be able to observe differences between these methods. We start with 2,000 sentences (34k constituents), and continue for 30 rounds until a total of 5,000 sentences has been sampled (ca. 87k constituents).

Method **two_stage** performs consistently better than both **random** and **entropy/unparsed** (Fig. 3). It reduces the amount of labelled data necessary to reach 85.5% f-measure by 32.6% as compared to **random** (Tab. 3). The central result of this paper is that, to reach this level, **two_stage** reduces the number of constituents by 8,800 constituents against the state-of-the-art method **entropy/unparsed**: a reduction by a further 10.1%. Also, it has consistently higher precision and recall than **entropy/unparsed** after the labelling of 80k constituents (Tab. 4).

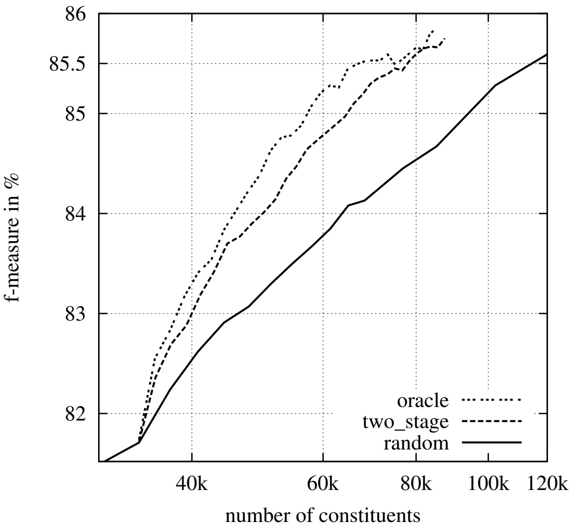

Figure 5: Comparison of f-measure learning curves. Number of constituents is given on a log scale.

Method	Cov	Prec	Rec	Fm
random	99.0	85.1	84.0	84.5
two_stage	99.7	86.0	85.2	85.6
entropy/unparsed	99.6	85.7	84.9	85.3

Table 4: Parseval values for different metrics after 80,000 constituents have been annotated.

An ensemble method We now compare performance of the new method against a state-of-the-art ensemble method, namely the KL-divergence to the mean for an ensemble of n bagged parsers, cf. Subsec. 2. The method **kl-div/unparsed** preferably selects unparsed pool sentences, and fills the batch with sentences having a high mean KL-divergence. (We consider a sentence as unparsed when at least one of the ensemble members fails to deliver an analysis.) As in the previous experiment, we start with 2,000 sentences and continue for 30 rounds. We set the ensemble size to $n = 5$.

Fig. 4 shows ensemble method **kl-div/unparsed** to perform better than **two_stage** until 57k constituents. After this point, **two_stage** performs clearly better than **kl-div/unparsed**. This is actually a surprising result, given that both methods perform similar jobs: They select unparsable sentences according to one or more bagged parsers, and then apply an information-theoretic measure, either entropy or KL-divergence. We conjecture that, after having filtered the difficult examples in the first stage, the second stage should make use of the information to be had from a fully trained parser. At any rate, our proposed method is conceptually simpler and also quicker to compute than an ensemble-based method.

6 Understanding the New Method

Acquiring the annotation of objectively difficult sentences should improve the parser. We employ an oracle-based experiment to test this claim. Method **oracle** selects sentences whose preferred parse tree (according to the current grammar) has low f-measure as determined against a gold-standard tree. Fig. 5 shows that method **oracle** performs consistently better than our best result, the new **two_stage** method. This suggests that a selection method which successfully targets difficult sentences (low f-measure) will perform well.

	Sentences	F-measure	Pearson
parsable	4,919	83.9%	-0.37
unreliable	112	71.5%	0.05

Table 5: Average f-measure and correlation coefficients between entropy and f-measure

In another experiment, we train the parser on a randomly sampled training set of 2,000 sentences, and apply it to a test set of 5,000 sentences. We are interested in the degree of correlation between the variables f-measure (preferred tree against gold-standard tree) and tree entropy. A correlation analysis over all 4,919 **parsable** sentences shows that the two variables are indeed (negatively) correlated, cf. Tab. 5. Pearson's coefficient is -0.37. Given the size of the considered data set, correlation is highly significant ($p = 0.01$). Selection according to entropy will thus tend to pick low f-measure sentences. Given the observation from the oracle experiment that it is beneficial to target low f-measure sentences, this finding explains why entropy is a useful selection method.

If we now apply a bagged version of the parser to the same test set, more sentences become unparsable since we eliminate some infrequent parse events. Focusing on the 112 **unreliable** sentences which are parsable under a fully trained model, but not under a bagged model, we find a Pearson coefficient between entropy and f-measure close to 0 (Tab. 5). In other words, entropy and f-measure are uncorrelated, and entropy cannot reliably select difficult examples within this class of sentences. What is more, the average f-measure within these **unreliable** sentences is more than 12 percentage points below average, indicating that acquiring their true parse trees will be particularly useful.

Note that the first stage of our new method targets exactly these kind of unreliable sentences. The above experiments demonstrate why the new method is indeed successful.

7 Related Work

Preferably selecting high entropy examples has been shown to be an effective method for parsing [Hwa, 2000]. Selecting unparsed sentences has been previously suggested because of their high uncertainty, e.g. in [Thompson *et al.*, 1999]. However, to the best of our knowledge, this effect has not been quantified before. Bagging ensemble members (or alternatively random perturbation of event counts) has been explored in the context of active learning by [Argamon-Engelson and Dagan, 1999; McCallum and Nigam, 1998]. In particular, Argamon-Engelson and Dagan have indicated that these methods target low frequency events. Bagging (and boosting) a parser ensemble has been employed to increase parser performance [Henderson and Brill, 2000]. However, the application of a bagged parser ensemble to active learning is new. Density estimation has been suggested as a method to guard against selecting of outliers, e.g [McCallum and Nigam, 1998; Tang *et al.*, 2002]. This approach is orthogonal to our suggested new method, and combining the two may well result in even better performance.

8 Conclusion

We demonstrated a number of points in this paper. First, we investigated the effect of targeting unparsed sentences. This is a simple, but very effective way to increase labelled recall and thereby f-measure. This method has been used implicitly before, but to our knowledge the effect of this strategy has not been quantified previously. Secondly, we presented a novel, two-stage method which particularly targets sentences which cannot be reliably selected using popular active learning methods. We showed that the proposed method works better than uncertainty sampling alone. Also it compares favourably against a state-of-the-art ensemble method based on bagging. Finally, an oracle-based experiment indicated that targeting (objectively) difficult sentences is a good strategy. Furthermore, we demonstrated that entropy and f-measure are significantly correlated in general. However, they are uncorrelated for exactly the class of sentences of which our new method takes care. This explains why the new two-stage method performs well. In future work, we would like to investigate if the proposed two-stage method can be applied to applications other than parsing.

Acknowledgments

We would like to thank James Henderson, Mirella Lapata, and Andrew Smith for valuable discussions and comments on this work.

References

[Argamon-Engelson and Dagan, 1999] Shlomo Argamon-Engelson and Ido Dagan. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360, 1999.

[Bikel, 2004] Daniel M. Bikel. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511, 2004.

[Black *et al.*, 1991] E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings, Speech and Natural Language Workshop*, 1991.

[Brants, 2000] Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference*, 2000.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–14, 1996.

[Charniak, 2000] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, 2000.

[Collins, 1997] Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[Henderson and Brill, 2000] John C. Henderson and Eric Brill. Bagging and boosting a treebank parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, 2000.

[Hwa, 2000] Rebecca Hwa. Sample selection for statistical grammar induction. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 45–52, 2000.

[Hwa, 2001] Rebecca Hwa. *Learning Probabilistic Lexicalized Grammars for Natural Language Processing*. PhD thesis, Harvard University, 2001.

[Lewis and Gale, 1994] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[McCallum and Nigam, 1998] Andrew McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 350–358, 1998.

[Pereira *et al.*, 1993] Fernando C.N. Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993.

[Seung *et al.*, 1992] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[Tang *et al.*, 2002] M. Tang, X. Luo, and S. Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 120–127, 2002.

[Thompson *et al.*, 1999] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the 16th International Conference on Machine Learning*, pages 406–414, 1999.