

Active Learning for Dependency Parsing by A Committee of Parsers

Saeed Majidi

Department of Computer Science
Tufts University
Medford, MA USA
saeed.majidi@tufts.edu

Gregory Crane

Perseus Project
Tufts University
Medford, MA USA
gregory.crane@tufts.edu

Abstract

Data-driven dependency parsers need a large annotated corpus to learn how to generate dependency graph of a given sentence. But annotations on structured corpora are expensive to collect and requires a labor intensive task. Active learning is a machine learning approach that allows only informative examples to be selected for annotation and is usually used when the number of annotated data is abundant and acquisition of more labeled data is expensive. We will provide a novel framework in which a committee of dependency parsers collaborate to improve their efficiency using active learning techniques. Queries are made up only from uncertain tokens, and the annotations of the remaining tokens of selected sentences are voted among committee members.

1 Introduction

Emerging digital libraries must manage not only surrogates for traditional publications (such as PDF and HTML files) but the data-sets associated with, and sometimes derived from, these traditional publications. Linguistic annotations such as part of speech and syntactic function of the words are increasingly important for students of language. These annotations are, however, expensive to collect in the best case when native speakers are readily available. Developing such databases of annotations for historical languages, where no native speakers are alive and where few have even developed advanced language skills; the process becomes even more expensive. This paper describes how such linguistic data can be extracted automatically and lays the foundations for smart digital libraries that can not only offer preservation and access, but also generate fundamental data.

In the occasions that the number of annotated data is abundant and acquisition of more labeled data is expensive, using active learning techniques is one of the promising approaches. Active learning is a machine learning approach that allows only informative examples to be selected for labeling. The main idea of utilizing active learning is that the learner can achieve better performance with fewer training data if it can choose the examples it needs to learn from in an intelligent manner.

Active learning has been successfully applied to many of natural language processing applications such as information extraction (Jones et al., 2003; Culotta et al., 2006), named entity recognition (Kim et al., 2006; Velachos, 2006; Laws and Schütze, 2008), part of speech tagging (Argamon-Engelson and Dagan, 1999; Ringger et al., 2007), text classification (Schohn and Cohn, 2000; Tong and Koller, 2002; Hoi et al., 2006), word segmentation (Sassano, 2002) and word-sense disambiguation (Zhu and Hovy, 2007).

In this paper we investigate active learning techniques that can be used for dependency parsing to help us reach better performance with cheaper annotation cost. The main motivation of this work is, as McDonald and Nivre showed in (McDonald and Nivre, 2007), different parsers generate different models that produce different types of errors. Hence, when two or more parsers agree about annotation of a token it is not worth to spend a budget to know its true annotation. We will provide a novel framework in which a committee of dependency parsers collaborate to improve their efficiency using active learning techniques. In each round of annotation, the committee of parsers select a few tokens they disagree most from uncertain selected sentences. An expert then annotates the chosen tokens, and the committee members vote about

the annotations of the rest of the tokens of the selected sentences.

The rest of this paper is structured as follows. We briefly review active learning and pool-based sampling as a commonly used framework of active learning in section 2. We also talk about different querying scenarios for pool-based active learning. In section 3, we bring a literature survey of applications of active learning for the task of dependency parsing. We also introduce an uncertainty sampling framework with a different confidence measure as a baseline to compare and evaluate our methods. We introduce our proposed methods of committee-based active learning for dependency parsing in section 4. The experimental results are presented in section 5 and show that we can gain better results using a committee-based strategy. We conclude this paper at section 6 with a brief summary of the findings and an outline of the future work.

2 Active Learning

Active learning is one of the mostly used applied machine learning techniques. In an active learning framework, the learner is able to ask an oracle, a domain expert, about the annotations of only those unlabeled instances that could help it to improve itself.

Pool-based sampling (Lewis and Gale, 1994) is one of the basic scenarios for active learning. It is usually used when there is a small set of labeled data \mathcal{L} and a large pool of unlabeled data \mathcal{U} . In each round, instances are chosen from \mathcal{U} according to a selection strategy, and their labels are asked from an oracle. Then, the labeled instances are added to \mathcal{L} . This process is repeated until there are enough annotated data at \mathcal{L} .

At the heart of each active learning scenario there is a *query strategy* that formulates how the next instances should be selected to be queried for their labels. There are different query strategies in the literature. Two of commonly used query strategies are *uncertainty sampling* and *query-by-committee*.

The simplest and most commonly used method for selecting samples to be annotated is uncertainty sampling (Lewis and Gale, 1994). In uncertainty sampling, after training a model using existing labeled data in \mathcal{L} , it is used to predict the labels of instances in \mathcal{U} . Then the learner selects the instances about whose labels it is less confident.

Another approach for sample selection is query-by-committee (Sebastian Seung and Sompolinsky, 1992) (QBC). In a QBC framework, the system contains a committee of c competing models $M = \theta^1, \theta^2, \dots, \theta^c$ all trained on the current labeled data set \mathcal{L} . Then each of the committee members votes on the label of examples in \mathcal{U} . The query contains the samples about which committee members most disagree.

We refer the interested reader to look at Settels (Settels, 2009) for a comprehensive survey of active learning in general and to Olsson (Olsson, 2009) for a literature survey of active learning in the context of natural language processing.

3 Active Learning for Dependency Parsing

There are two common approaches for the selection of samples to be queried: *sentence-wise* sample selection and *token-wise* sample selection.

In sentence-wise sample selection, the learner selects full sentences that it is not confident about their parsing to be annotated. Tang et al. (Tang et al., 2002) and Hwa (Hwa, 2004) use sentence entropy, calculated based on the n -best parses of a sentence S , to measure the uncertainty of each sentence. The k uncertain sentences are selected for the annotation. Lynn et al. (Lynn et al., 2012) use QBC-based active learning for bootstrapping a dependency bank at the sentence level.

Another approach is to select only some parts of a sentence for the annotation rather than full sentences. Sassano and Kurohashi (Sassano and Kurohashi, 2010) use the score that parser assigns to dependencies to select uncertain dependencies for the task of Japanese dependency parsing. They utilize syntactic constraints of Japanese to decide about annotations of the rest of dependencies. Mirroshandel and Nasr (Mirroshandel and Nasr, 2011) use a combined active learning strategy to select uncertain tokens for the annotation. They first calculate each sentence entropy based on the n -best scores that the parser generates for each sentence. After selecting k most uncertain sentences, they calculate the attachment entropy for every token of each sentence. Given the n -best parse of a sentence S , they compute the attachment entropy of token w based on the number of its possible governors in the n -best parse. They use parser annotations for the rest of tokens of selected sentences.

We use the same idea of selecting uncertain sentences first and then choosing some parts of those sentences for the annotation. But the contribution of our work is that unlike (Mirroshandel and Nasr, 2011) work, we use a committee of parsers rather than only one parser. The parsers of committee collaborate with each other to choose the next set of tokens to be queried, and decide about the annotation of remaining set of tokens. We also use a different query method and uncertainty measure to select the tokens. In this paper we extend our approach in (Majidi and Crane, 2013). We provide a solid algorithm and investigate the effect of number of selected tokens for query.

3.1 Uncertainty Sampling Word Selection

We set *uncertainty sampling word selection* as a baseline to compare our work with it. The confidence measure that we use for uncertainty sampling is KD-Fix, K -draws with fixed standard deviation, proposed in (Mejer and Crammer, 2011). KD-Fix is a stochastic method to generate K alternatives for the best labeling. Given a model parameter μ learned by the parser, a Gaussian probability distribution with an isotropic covariance matrix, $\Sigma = sI$, is defined over that, $w \sim \mathcal{N}(\mu, \Sigma)$. Then a set of K weight vectors w_i are drawn, and each one outputs a single alternative labeling. If $y_w^{(i)}$, $i = 1, \dots, K$ be the K alternative labeling generated for token w and \hat{y}_w be the actual predicted label by the model, the confidence in the label \hat{y}_w is defined as :

$$\nu_w = \frac{|\{i : \hat{y}_w = y_w^{(i)}\}|}{K}$$

Along parsing the sentences in \mathcal{U} we have the parser to generate the confidence score for each edge that shows the parser confidence on the correctness of that edge (token). We assign the confidence score of each sentence as the average confidence score of its tokens. Then we rank the sentences of \mathcal{U} based on the computed confidence score and select k sentences with least confidence score. For each of these k sentences we choose l tokens with the least confidence score among the tokens of that sentence. We ask the expert to annotate the selected tokens (head and relation), and the trained parser annotates rest of the tokens. Finally these k annotated sentences are added to \mathcal{L} .

4 Committee-Based Active Learning for Dependency Parsing

We use a committee of c parsers in the active learning framework. Each parser is first trained, using labeled pool \mathcal{L} , and then predicts the head and the relation to the head of every instance in unlabeled pool \mathcal{U} . We show the head prediction of parser P_i for token w as $h(P_i, w)$, and its relation prediction as $r(P_i, w)$. Here we assume that the head and relation of token w are predicted independently. The trained parsers parse a separate test set \mathcal{T} and their parsing accuracy on the test set is computed. $UA(P_i)$ shows the unlabeled accuracy of parser P_i and $LA(P_i)$ shows its labeled accuracy. *Unlabeled accuracy* is the percentage of correct head dependencies, and *labeled accuracy* is the percentage of correct relations for the correct dependencies that the parser predicts.

To select the next tokens to be included in the query, two entropy measures, $HE(w)$ and $RE(w)$, are computed as the confidence score of each token w :

$$HE(w) = - \sum \frac{V(h_j, w)}{\sum_{i=1}^c UA(P_i)} \cdot \log \frac{V(h_j, w)}{\sum_{i=1}^c UA(P_i)}$$

$HE(w)$ is the head vote entropy of token w in which h_j varies over all of the head values assigned to token w and $V(h_j, w)$ is the number of votes assigned to h_j as the head of token w :

$$V(h_j, w) = \sum_{\forall i, h_j = h(P_i, w)} UA(P_i)$$

In a same way, we calculate the relation vote entropy for token w :

$$RE(w) = - \sum \frac{U(r_j, w)}{\sum_{i=1}^c LA(P_i)} \cdot \log \frac{U(r_j, w)}{\sum_{i=1}^c LA(P_i)}$$

$$U(r_j, w) = \sum_{\forall i, r_j = r(P_i, w)} LA(P_i)$$

$RE(w)$ is the relation vote entropy of token w and r_j varies over all of the relations assigned to token w . $U(r_j, w)$ is the number of votes assigned to r_j as the relation of token w to its head.

The entropy measure of token w is computed as the mean value of the head entropy and relation entropy:

$$WE(w) = \frac{HE(w) + RE(w)}{2}$$

Finally, for a sentence S we assign the average entropy value of all its tokens as the sentence entropy:

$$SE(S) = \sum_{i=1}^n \frac{WE(w_i)}{n}$$

4.1 Single Parser Prediction

After computing the confidence score of each token and sentence in \mathcal{U} , we select k sentences with most entropy value. For each sentence we ask the expert to annotate l tokens that has the highest entropy. Here, we select one of the parsers as the main parser and use it to annotate the rest of the words of those k selected sentences and add them all to \mathcal{L} .

4.2 Weighted Committee Prediction

In this section we use the same querying strategy as the previous one to select l tokens of k sentences with higher entropy value to be annotated by the expert. But for predicting the labels of the rest of the words, we run a majority voting among the parsers. The vote of each parser P_i for predicting the head and relation of each token is weighted by its labeled and unlabeled accuracy, $LA(P_i)$ and $UA(P_i)$, and the label of each token is set to the one that has majority of votes among parsers. Algorithm 1 shows the QBC active learning for dependency parser.

5 Experiments

To evaluate the proposed method, we set up 5 different experiments. In the first two ones, we select the tokens that should be annotated randomly. In one case we first select sentences randomly from unlabeled pool, and then we choose some random tokens of each selected sentence and ask for their labels. The trained parser annotates the rest of the words. In another case, we have a committee of parsers that vote about the annotations of the rest of the words of sentences that should be added to the labeled pool. In the third experiment, we try the query by uncertainty sampling with the confidence measure that we discussed earlier in 3.1. We set up these three experiments as baselines with which we compare the proposed method. The last two experiments are related to the QBC active learning. In one case we use only one of the parsers of the committee to predict the rest of the tokens of selected sentences that we do not choose for expert

Algorithm 1 QBC Active Learning for Dependency Parsing with Committee Prediction

```

 $\mathcal{L} \leftarrow$  Initial labeled training set
 $\mathcal{U} \leftarrow$  Unlabeled pool
 $\mathcal{T} \leftarrow$  Test set
 $C \leftarrow P_1, \dots, P_c$  // A committee of  $c$  parsers
repeat
   $W \leftarrow []$  // Weight vector
  for  $\forall P_i \in C$  do
     $P_i \leftarrow \text{train}(\mathcal{L})$ 
     $\mathcal{U}'_i \leftarrow \text{parse}(\mathcal{U}, P_i)$ 
     $\mathcal{T}'_i \leftarrow \text{parse}(\mathcal{T}, P_i)$ 
     $w_i \leftarrow$  Calculate weight of  $P_i$  given  $\mathcal{T}$  and  $\mathcal{T}'_i$ 
     $W \leftarrow W \cup w_i$ 
  Calculate confidence score of each token of  $\mathcal{U}'$ 
  given  $W$ 
   $\mathcal{S} \leftarrow k$  least confident sentences of  $\mathcal{U}'$ 
   $\mathcal{I} \leftarrow \{\}$ 
  for  $\forall s \in \mathcal{S}$  do
    Query the expert  $l$  least confident tokens of  $s$ 
    Vote among parsers for annotations of the rest
    of tokens of  $s$ 
    Add new annotated sentence to  $\mathcal{I}$ 
   $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{I}$ 
   $\mathcal{U} \leftarrow \mathcal{U} - \mathcal{S}$ 
until  $\mathcal{U}$  is empty or some stopping criteria is not
met
return  $C, \mathcal{L}$ 

```

annotation. The latter case is an implementation of algorithm 1. All parsers of committee decide together about the annotation of those tokens. The votes of each parser are weighted by its accuracy on the test set.

We use MSTParser (McDonald et al., 2005) as the main parser in each experiment that only needs one parser. To set up the QBC experiments, we make a committee of three parsers including MSTParser, Mate parser (Bohnet, 2010) and DeSR parser (Attardi, 2006).

5.1 Corpora

In our experiments we use data sets from Ancient Greek Dependency Treebank (Bamman et al., 2009) and TUT corpora (Bosco et al., 2000). The Ancient Greek Dependency Treebank (AGDT) is a dependency-based treebank of literary works of the

Data set	Language	# Of Sentences	# Of Tokens	Avg Sent Length
Sophocles	Ancient Greek	3307	39891	12.06
Wiki	Italian	459	14747	32.12

Table 1: Corpora used in the experiments.

Archaic and Classical age published by Perseus. It includes 13 texts of five Greek scholars from which we select Sophocles’ works. TUT corpora has been organized in five sections. Here we use the Wiki section that includes 459 sentences, randomly chosen from the Italian version of the Wikipedia. Table 1 shows the number of sentences and tokens for both data sets.

5.2 Experimental Setup

In each experiment we divide whole sentences of a text in two random training and test sets. Training set has 75% of the sentences of the text. We also divide the training set to two pools \mathcal{L} and \mathcal{U} . Initially we put 10% of training data in \mathcal{L} as the labeled pool and the rest go to unlabeled pool \mathcal{U} . In each iteration we select 10% of unlabeled sentences in initial training set from \mathcal{U} and after having their annotations add them to \mathcal{L} . For every text, we replicate each experiment for 10 different random seeds.

5.3 Experimental Results

Figure 1 shows the learning curve for unlabeled dependency parsing of the Wiki data set when 10 tokens per sentence are selected for annotation. x arrow grows with the number of tokens in training set, and y arrow shows the test set accuracy¹ that the main parser, MSTParser, can achieve². We can observe that the methods which use an active query strategy do a better job than those methods which are based on random selection strategy. Among active learning methods, QBC strategy works better than the rest. One explanation could be the way that the queries are selected. Since each parser generates a different model, they can make different types of errors. The selected query by the committee is the one that most of the members have difficulty on that, and hence knowing its label is more informative. We run one-tailed, paired t-tests to test if these differences are statistically

¹The percentage of correct head dependency predictions

²Our experiments on Sophocles show the same behavior. But due to the lack of space we do not report them here.

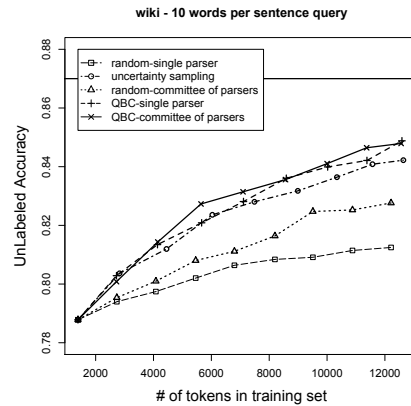


Figure 1: Learning curves for dependency parsing of the Wiki data set. 10 words per sentence are selected. The solid horizontal line is the unlabeled accuracy of parser on the fully annotated training set.

significant. The t-tests run for the best performance that each of the methods can achieve after the final loop of active learning. Table 2 shows the p-values for the case that 10 tokens of each sentence is selected.

Method	p -value
Random single parser	0.0014
Random committee of parsers	0.03
Uncertainty sampling	0.002
QBC single parser	0.0006

Table 2: p -values of paired t-tests to compare QBC-committee of parsers with other methods.

The number of selected tokens, variable l in algorithm 1, has a direct effect on the performance that we get. To investigate how many tokens we need to select, we plot the learning curves for both the Wiki and Sophocles when different number of tokens have been selected. Figure 2 shows the learning curves of unlabeled dependency parsing for Wiki and Sophocles. It compares QBC and random selection scenarios when 1, 5, and 10 words are selected. Solid lines depict QBC strategy and dashed ones show random selec-

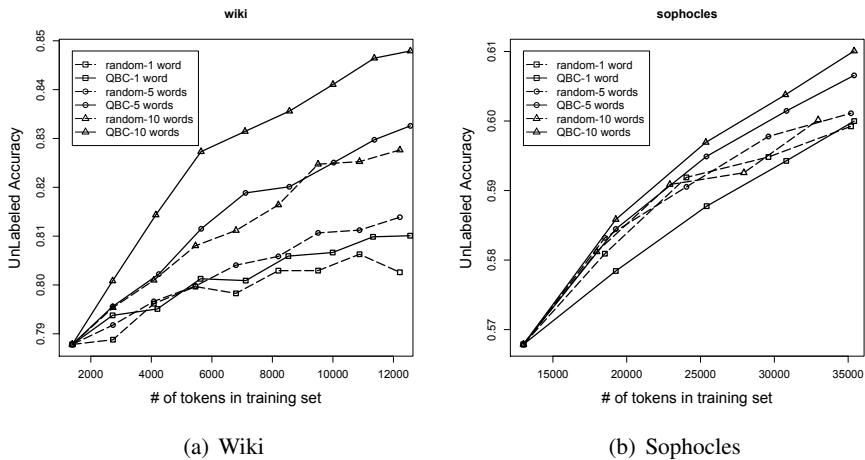


Figure 2: Comparing learning curves of unlabeled dependency parsing for QBC active learning (solid lines) and random selection (dashed lines) for different number of selected tokens per sentence (1, 5, and 10).

tion. One can see that when we only select one token per sentence, both active learning and random selection strategies perform almost the same. When we increase the number of selected tokens to 5 and then 10, we observe that for Sophocles active learning approach can achieve better than random selection. For the Wiki, selecting 10 tokens randomly is almost the same as selecting 5 tokens with active learning.

One reason could be the length of the sentences in each data set. As we can see in table 1, the average sentence length in the Wiki is as twice as the average sentence length of Sophocles. Table 3 reports the percent of the expert’s annotated tokens that we have in the training set after the final loop of active learning. When only 1 token per sentence is selected in each round, only less than 10% of total tokens in final training set have gold standard label. Therefore one should not expect that the active learning approach performs better than random selection. For the Wiki data set that has long sentences, 32 tokens per sentence in average, when 5 tokens from each uncertain sentence are selected, we finally reach a point that only 15% of tokens have gold standard label and the random selection of 10 tokens is doing better than that. But as Sophocles has smaller number of tokens per sentence, 12 tokens per sentence in average, selecting 5 uncertain tokens from each sentence will lead us to a point that finally more than 40% of tokens in the training set have gold standard label, and hence active learning has better performance even better than the case

Data set	1-token	5-token	10-token
Sophocles	8%	41%	83%
Wiki	3%	15%	31%

Table 3: Percentage of tokens in the final training set annotated by an expert.

that 10 tokens per sentence are selected randomly.

6 Conclusions and Future Work

We have set up an active learning framework with a committee of dependency parsers. The experimental results show that using a committee of parsers, we can reach better accuracy with less cost of annotation than the case where there is only one parser with uncertainty sampling.

We are currently working on a model that instead of a single oracle we have a committee of experts with different levels of expertise. We want to build a model to combine the annotation of those experts together and send that feedback for the parser.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0910165 and by a Tufts Collaborates 2011 seed grant. Thanks are also due to Bridget Almas and Alison Babeu for their invaluable research assistance.

References

- Shlomo Argamon-Engelson and Ido Dagan. 1999. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360.
- Giuseppe Attardi. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 166–170. Association for Computational Linguistics.
- David Bamman, Francesco Mambrini, and Gregory Crane. 2009. An ownership model of annotation: The ancient greek dependency treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories*, pages 5–15.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97. Coling 2010 Organizing Committee, August.
- Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. 2000. Building a treebank for italian: a data-driven annotation schema. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 99–105.
- Aron Culotta, Trausti Kristjánsson, Andrew McCallum, and Paul Viola. 2006. Corrective feedback and persistent learning for information extraction. *Artificial Intelligence*, 170(14):1101–1122, October.
- Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. 2006. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642. ACM.
- Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276, September.
- Rosie Jones, Rayid Ghani, Tom Mitchell, and Ellen Riló. 2003. Active learning for information extraction with multiple view feature sets. In *the ECML workshop on Adaptive Text Extraction and Mining*.
- Seokhwan Kim, Yu Song, Kyungduk Kim, Jeong won Cha, and Gary Geunbae Lee. 2006. Mmr-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting*, pages 69–72. Association for Computational Linguistics.
- Florian Laws and Hinrich Schütze. 2008. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 465–472. Association for Computational Linguistics.
- David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. ACM/Springer.
- Teresa Lynn, Jennifer Foster, Elaine Ui Dhonnchadha, and Mark Dras. 2012. Active learning and the irish treebank. *The Annual Meeting of the Australasian Language Technology Association (ALTA 2012)*.
- Saeed Majidi and Gregory Crane. 2013. Committee-based active learning for dependency parsing. In *Research and Advanced Technology for Digital Libraries*, pages 442–445. Springer.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*. Association for Computational Linguistics, June.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.
- Avihai Mejer and Koby Crammer. 2011. Confidence estimation in structured prediction. *CoRR*, abs/1111.1386.
- Seyed Abolghasem Mirroshandel and Alexis Nasr. 2011. Active learning for dependency parsing using partially annotated sentences. In *the 12th International Conference on Parsing Technologies*, pages 140–149. Association for Computational Linguistics, October.
- Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing. Computer sciences technical report, Swedish Institute of Computer Science.
- Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108. Association for Computational Linguistics, June.
- Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for japanese dependency parsing. In *the 48th*

- Annual Meeting of the ACL*, pages 356–365. Association for Computational Linguistics, July.
- Manabu Sassano. 2002. An empirical study of active learning with support vector machines for japanese word segmentation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 505–512. Association for Computational Linguistics.
- Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 839–846.
- Manfred Oppel Sebastian Seung and Haim Sompolinsky. 1992. Query by committee. In *the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 287–294. ACM.
- Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Min Tang, Xaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *the 40th Annual Meeting of the Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics, July.
- Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March.
- Andreas Velachos. 2006. Active annotations. In *the ECML workshop on Adaptive Text Extraction and Mining*, pages 64–71. Association for Computational Linguistics.
- Jingbo Zhu and Eduard H Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, volume 7, pages 783–790.