Systematic Evaluation of Convergence Criteria in Iterative Training for NLP

Patricia Brent
Oak Ridge National Laboratory

Nathan Green
North Carolina State University
Oak Ridge National Laboratory

Paul Breimyer
North Carolina State University
Oak Ridge National Laboratory

Ramya Krishnamurthy
Oak Ridge National Laboratory

Nagiza F. Samatova
North Carolina State University
Oak Ridge National Laboratory
Corresponding Author: samatovan@ornl.gov

Abstract

Natural Language Processing (NLP) tasks, such as Named Entity Recognition (NER), involve an iterative process of model optimization to identify different types of words or semantic entities. This optimization to achieve a more precise model becomes computationally difficult as the number of iterations increase. The small datasets available for training typically limit the models. Adding iterations on such sets to further optimize the model can often cause over-fitting, which generally leads to reduced performance. Therefore, the choice of convergence criteria is a critical step in robust and accurate model building. We evaluate different convergence criteria in terms of their robustness, stopping threshold selection, and independence from the training data size and entity. The underlying framework employs a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) parameter optimization in the context of Conditional Random Fields (CRF). This paper presents a convergence criterion for robust training irrespective of semantic types and data sizes with two-orders of magnitude reduction in stopping threshold for improved model accuracy and faster convergence. Additionally, we examine convergence with active learning to further reduce the training data and training time.

Introduction

Convergence criterion is a critical issue for iterative machine learning tasks (Wallach 2002). The right choice of convergence criteria can reduce the time it takes to build the training model, while increasing the model performance. This paper systematically evaluates convergence criteria for a specific branch of machine learning, Natural Language Processing (NLP) of text, which is often unstructured and ambiguous in its meaning.

Named Entity Recognition (NER) techniques can assign structure to plain text by associating semantic meaning to noun phrases (or entities). Repeated exposure to tagged data and features describing different word types, or entities, aids in the construction of a computer model. The relationship of a feature to the given text determines the importance, or weight, in defining the correct tag. Based on the log-likelihood that the manually tagged words receive those tags given the specified weighted features, the process judges the model to be well-trained.

Training a model until its log-likelihood reaches zero (representing a 100% chance that the given words would be correctly labeled) is not only computationally impractical but is also likely undesirable. Particularly when dealing with small sets, a model may become so well fitted to the training data that its performance on outside data decreases. (See (Dietterich 1995) for further information on this problem, known as over-fitting.) Thus, implementing more expedient convergence criteria is necessary to stop the training process after a reasonable amount of time.

In this paper we systematically evaluate the log-likelihood function's norm, gradient norm, and relative change as convergence criteria for multiple entity types. For each criterion, we use L-BFGS (Liu & Nocedal 1989) for parameter optimization in a CRF model (Lafferty, McCallum, & Pereira 2001). Examples of evaluation metrics for the target convergence criteria include: (a) robustness vs. oscillation; (b) dependence vs. independence of convergence criteria on the training data size and the entity type; and (c) thresholds for stopping the convergence and their effects on model accuracy. Our findings, for instance, indicate that unlike the widely used 10^{-7} threshold value for stopping the convergence (Malouf 2002), a 10^{-5} stopping threshold produces similar F_1-measures of model accuracy while reducing training time. Likewise, our findings confirm that the gradient norm oscillates near extrema (Nocedal, Sartenaer, & Zhu 2002). We further expand this evaluation procedure to address the issue of reducing the amount of manual labor involved in tagging data required for model training. We use our active learning (AL) framework (Symons *et al.* 2006) for this purpose.

Background

Optimization Methods and Convergence Criteria

Two early comparative studies showed that CRFs train most efficiently using *numerical optimization* procedures. Malouf's results (Malouf 2002) suggest the superiority of first- or second-degree numerical optimization, particularly on sequential labeling tasks, such as those in NLP.

Quasi-Newton methods, popular among second-order numerical optimization algorithms, work with the Taylor series for the target function. The inverse of Hessian matrices is *approximated* iteratively in order to use curvature information, and line searches find the minimum or maximum of a given function. The most commonly used quasi-Newton methods is the one attributed to Broyden, Fletcher, Goldfarb, and Shanno (BFGS) (Liu & Nocedal 1989). The BFGS method has the added advantages of utilizing two-part approximate Hessians to better simulate second-order functions and converging without exact line searches. The Limited-Memory BFGS (L-BFGS) method is useful not only for its memory efficiency but also for its superior performance on problems with large numbers of variables (Liu & Nocedal 1989; Symons *et al.* 2006). Limited memory techniques store only a certain number of approximated inverse Hessians, which tends to improve performance even in cases where memory is not a problem.

However, the manner in which the point of convergence is determined strongly impacts how accurately an approximate method represents the desired result. Because the CRF is based on log-likelihood measurements, a logical metric for convergence is the *norm of the gradient* of the log-likelihood function, L_k taken at each iteration, k. The gradient is the vector of first derivatives in respect to the parameter θ of the optimization function. Its Euclidean norm is the magnitude of the gradient vector. One expects this number to be small near the optimum. When the gradient norm is smaller than this limit, the program stops and identifies the point as a stationary, or optimal, point. However, a recent study makes clear the unreliability of the gradient norm near convergence on certain datasets (Nocedal, Sartenaer, & Zhu 2002). Instead of steadily decreasing as it approaches the minimum, they show that the gradient tends to fluctuate back and forth, taking unnecessarily large amounts of time to reach the desired point. Despite having made a careful study of the convergence, Nocedal makes no recommendation for a superior criterion. Malouf (Malouf 2002) suggests the relative change in log-likelihood defined as $\Delta L = \frac{|L_k - L_{k-1}|}{L_k}$ as an alternative convergence criterion.

Ideally, the model performance on a set of test data would determine the convergence directly by the maximum, but the computational cost of testing the model at each step along the way is prohibitive for even a reasonably-sized dataset.

Conditional Random Fields

Conditional Random Fields (CRF), a type of Markov Random Field, are log-linear graphical models that typically use a global optimization routine to maximize the conditional log-likelihood of the model parameters given the available training data (Lafferty, McCallum, & Pereira 2001). They restrict the choice of a probability distribution to one that matches the independence assumptions explicit in the structure of a graph. In such a graph, the nodes are random variables, such as the positions in a sequence to which states (labels) will be assigned, and the edges represent the dependencies between nodes. Thus, the lack of an edge indicates conditional independence between the concerned nodes.

CRFs have quickly become one of the more popular supervised-learning models in the field of NLP. They are well suited for sequential labeling tasks, such as part-of-speech tagging, shallow parsing, and NER, outperforming many other popular model types on such problems. CRFs are known to have problems with over-fitting, a phenomenon in which attempting to reduce training error beyond a certain point can actually lead to reduced generalizability.

Active Learning

Active learning experiments seek to improve performance relative to the size of the training set in order to reduce global training time, human labor, and computation time. One way to increase the ratio is to selectively reduce the size of the training set, leaving only the best examples. Existing suggestions to this end include measurements of uncertainty (Tong & Koller 2002), variance in prediction from query-by-committee (Freund *et al.* 1997), and risk minimization (Zhu, Lafferty, & Ghahramani 2003). However, uncertainty measures fail to take into account the data distribution, which can lead to the selection of outliers that negatively impact generalization performance. Alternate methods such as the ones suggested by Freund et al. (Freund *et al.* 1997) or McCallum and Nigam (McCallum & Nigam 1998), which do consider the data, show great improvement over simple uncertainty.

Because training is a computationally expensive process, active learning is often performed in batch by adding sets of samples simultaneously. Techniques for choosing representative batches have become popular; for example, Brinker (Brinker 2003) and Shen et. al (Shen *et al.* 2004) suggest a geometric distance method for Support Vector Machines (SVMs). Shen's work actually combines uncertainty, representativeness, and diversity for improved data selection (Shen *et al.* 2004). In general, attempts to use representative data, particularly with batch selection, have led to performance improvements relative to dataset size.

Our framework (Symons *et al.* 2006) uses multi-criterion active learning in CRFs to identify a small but sufficient set of text samples to train CRFs. The empirical results demonstrated that this approach could reduce manual annotation costs beyond using model uncertainty alone, which is particularly encouraging given the difficulty of obtaining new training data when applying models to new concepts and domains in NLP tasks. For the NER task, the framework reduced the training set from 8,646 to 1,150 sentences while maintaining a comparable F_1-measure. Similar studies have shown Active Learning techniques to reduce the size of training data as well (Brinker 2003; Shen *et al.* 2004; Sassano 2002). Choosing the training dataset selectively improved the generalization performance of trained CRF models. Thus, active learning is a potential method for dealing with over-fitting in CRFs that can enhance the results of other known methods.

Methodology

Architecture and Implementation

Any machine learning system that involves humans selecting and tagging the data on which to train a model typically contains two major iterative cycles: the internal, or *Training,* cycle to train the model given a set of tagged data, and the external, or *Active Learning*, cycle to control which data is sent to be trained (Figure 1). For the *Training* cycle, various choices at multiple levels are made, as depicted in Figure 2. For our implementation, we used the CRF at the maximum entropy model level, the L-BFGS method at the parameter

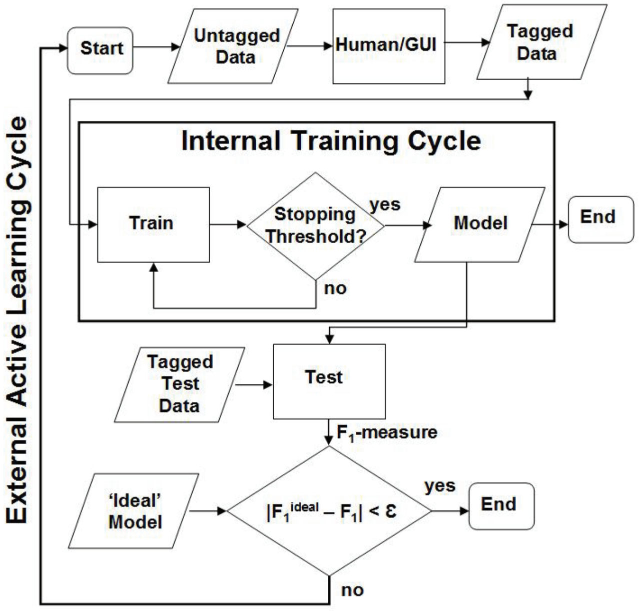

Figure 1: **General process flow of a two-cycle active learning training system.**

optimization level, various measures at the convergence criterion level, and various thresholds for stopping the training at the threshold selection level. We selected the L-BFGS method to optimize the parameters for the maximum log-likelihood function due to its benefits in memory and computation time on large-scale datasets. Our evaluation experiments on convergence criteria aim to find the 'best' convergence criterion and the 'best' stopping threshold for this cycle.

We used the open-source CRF package (Sarawagi 2005), which considered convergence according to the log-likelihood's gradient norm. By default, L-BFGS considers convergence according to the ratio of the gradient norm to the function norm.

Our active learning software (Symons *et al.* 2006) controlled the external, or *Active Learning*, cycle, which selects data that our model is most uncertain about. Therefore, correctly tagging this data by users would most benefit the model.

Evaluation Methodology

Internal Training Cycle Evaluation. Our experiments aim to determine robust criteria that terminate the internal training cycle with the least computation time and the best accuracy. One way is to preset the number of iterations, which may or may not be determined through testing. This is almost a guarantee for predictions with lower accuracy due to either too few iterations or over-fitting. Instead, we use a numerical value based on the log-likelihood function (i.e. its norm, gradient norm, relative change) between two consecutive iterations. Setting the desired stopping threshold for this convergence value to 10^{-7} is widely accepted. To remain consistent with previous studies, such as those by Malouf (Malouf 2002), we terminated the training when the convergence criterion attained this stopping value, or after 2000 iterations, if the latter occurred sooner. In our experiments we used three target convergence criteria: the relative change in log-likelihood, the Euclidean norm of the log-likelihood function, and the Euclidean norm of its gradient. The model performance, evaluated in terms of the F_1-measure, was checked every 100 iterations and at points where the other measures dropped below a certain power of 10. We also determined the minimum and maximum values obtained for each measure per 100 iterations in order to make observations about convergence trends.

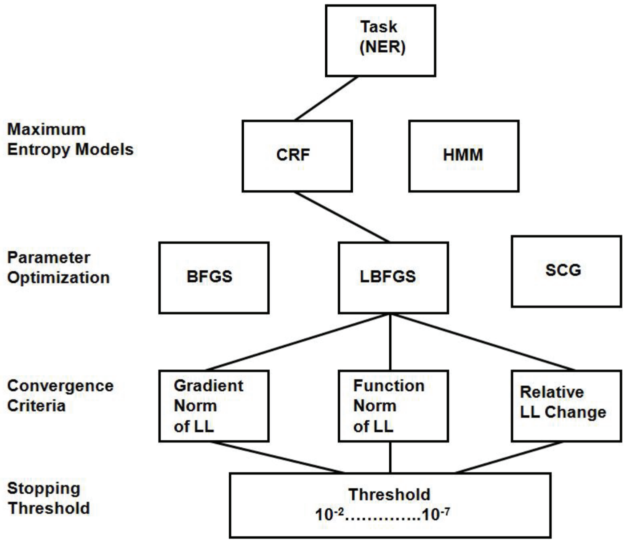

Figure 2: **Model of experimental decisions for the internal Training cycle. The lines connect the components used in the framework.**

External Active Learning Cycle Evaluation. To train our system with active learning (AL) we started with 100 randomly selected sentences to annotate. After each CRF training via L-BFGS, the active learning system selected another 50 sentences to annotate. We varied the convergence criterion (relative change in log-likelihood) in the training between 10^{-2} to 10^{-8}. Due to the randomness involved in picking the initial training sentences to annotate, this experiment was run 10 times and the reported results contain the average over the last 5 iterations and over all 10 experiments. For example, after 40 iterations of AL, the training data size reached 2050 sentences and we calculated the F_1-measure average for iterations 36 through 40 over the 10 experiments. We used the last 5 iterations along with 10 separate experiments to reduce the chance of fluctuations due to our sampling in AL. After each training run, the experiment evaluated the model's performance by using a separate test dataset from the Computational Natural Language Learning Conference (CoNLL).

Performance Evaluation Metric. To compare methods, we use the F_1-measure, the harmonic mean of recall and precision with an equal weight. Our goal is to find a convergence factor that consistently achieves a high percentage of

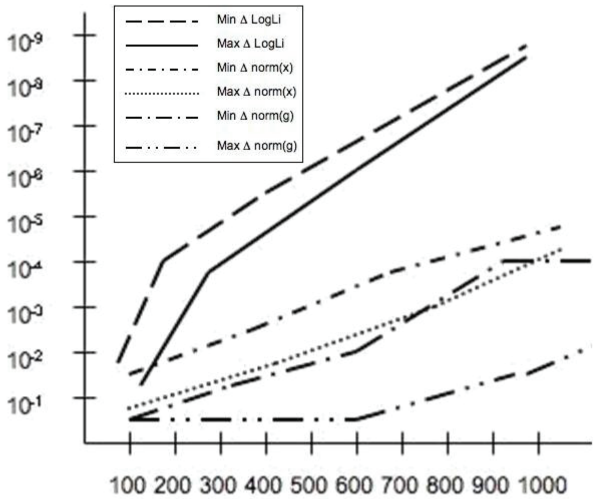

Figure 3: **Minimum vs. maximum relative rate of change (y-axis) of the various stopping measures over every 100 training iterations (x-axis).**

the maximum observed F_1-measure across all datasets and entities. We do not expect to increase previously observed F_1-measures but to show computation time reduction with comparable F_1-measure.

Data Sets. We use entities for people, organizations, and locations obtained from CoNLL and evaluated methods using datasets ranging from 50 to 4,000 sentences to train our models; the remaining sentences are used as a test set, and then we vary the convergence criteria.

Results

Training Cycle Convergence

In this section, we report our evaluation results and address the following questions: (a) which target convergence criteria under study are robust; (b) whether the optimum threshold for stopping at convergence depends on the type of the named entity, and (c) whether the threshold depends on the size of the training data. The following three convergence criteria are evaluated: function norm (or *norm(x)*) of log-likelihood, gradient norm (or *norm(g)*) of log-likelihood, and relative change (or Δ*LogLi*) in log-likelihood. The following three entity types are used: *Location*, *Person*, and *Organization*.

We first trained the model using the *default* L-BFGS convergence criterion defined as *the ratio of the gradient norm to the function norm*. The default stopping threshold for this ratio is 10^{-7}. The training terminated after 2000 iterations if it had not previously stopped by reaching the stopping threshold for its convergence ratio. The F_1- measures attained at this *default convergence* for *Location*, *Person*, and *Organization* were 0.922772, 0.84936, and 0.774273, respectively.

Each model was then trained for every named entity and every target convergence criterion until one of the two stopping conditions was satisfied: the 10^{-7} threshold for the convergence criterion or 2000 iterations. In terms of the F_1-measure, we noted that our experiments converged to the default threshold long before training concluded. Therefore, results obtained from the internal training cycle confirmed the usefulness of more expedient convergence criteria to achieve high F_1-measures more quickly.

F_1-measure per entity type			
Δ*LogLi*	**Location**	**Person**	**Organization**
10^{-4}	**0.922816**	0.838222	0.763786
10^{-5}	**0.923077**	0.847694	**0.779285**
10^{-6}	0.922772	**0.849450**	**0.774405**
10^{-7}	0.922772	**0.849609**	**0.775226**
10^{-8}	N/A	0.849029	0.773956

Table 1: Low change in log-likelihood predicts convergence-quality or higher performance across all three entities. Bold typeface indicates F_1-measures higher than those at convergence; N/A is reported for criteria not reached before training stopped.

Robustness of Convergence Criteria. Figure 3 depicts the relative change of the minimum and maximum values over 100 iterations for every target convergence criterion using *Organization* as the selected entity type. The log-likelihood has a relative change of less than 10^{-5} after iteration 500. Yet, the gradient norm oscillates by two orders of magnitude between its maximum and minimum values even after iteration 800. Although the function norm is not as oscillatory as the gradient norm, its maximum change per iteration is at least two orders of magnitude worse than the log-likelihood change after iteration 500, reaching 4 orders of magnitude difference after iteration 800. For the *Location* entity, the iteration to iteration change in function norm dropped below 10^{-4} around the time of convergence, but for *Person* and *Organization*, the change in this norm remained large even after convergence (data not shown). The gradient norm continued to vary greatly from iteration to iteration even up to convergence for all three entity types. For these reasons, we eliminated both function and gradient norms from further analysis as possible convergence criteria.

Independence of Threshold Selection on Entity Type. Given relative log-likelihood change as our target convergence criterion, the next question is whether to select an optimum stopping threshold uniformly defined across different entities, or for each individual entity type. Our experiments for three target entities demonstrated that the relative log-likelihood change consistently dropped below 10^{-5} near convergence in all three entities (see Table 1). F_1-measures from the models that first obtained this low shift in log-likelihood either very closely approximated the results after training ended (that is, within 0.1%), or exceeded the performance observed at convergence in several cases. This slight drop in performance near convergence may point to overfitting the training data.

Independence of Threshold Selection on Training Data Size. Having thus established the superiority of relative change in log-likelihood over the other two criteria, irre-

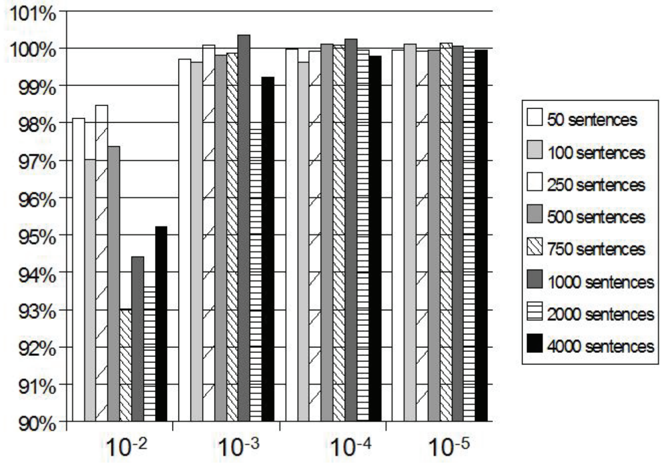

Figure 4: **Average percentage of final F_1-measure (y-axis) attained by each model for a given dataset size at different threshold values (x-axis).**

spective of the entity chosen, we next determined whether varying the amount of training data will affect our convergence criteria. Using randomly selected sets of 100, 250, 500, 750, 1000, 2000, and 4000 sentences from the CoNLL corpus, models were trained to recognize the *Location* entity and then tested whenever they reached a certain log-likelihood value. The results, shown in Figure 4, point toward a log-likelihood of 10^{-4} or 10^{-5} being most effective. Some results in Figure 4 achieve scores above 100% showing that looser convergence criteria can have greater performance since these results are relative F_1-measures compared against results using 10^{-7} as the convergence criteria. The relative performance of small datasets appears to outperform larger datasets for 10^{-2} , however, this is due to the small datasets' lower F_1-measure at the 10^{-7} benchmark compared to the larger datasets. For some sizes, 10^{-4} appears to produce better results, but given the random nature of the datasets selected, this result is not definitive. Larger datasets appear to require a more accurate log-likelihood for high performance. Therefore, using a 10^{-5} stopping threshold seems to be better because almost every model, regardless of dataset size, reached 99% (or more) of the F_1-measure.

Active Learning Cycle Convergence Evaluation

With relative change in log-likelihood shown to be the proper convergence criterion, and this criterion shown to be independent of the training data size, we questioned whether active learning (AL) could further reduce computation time and human effort.

F_1-measures attained at a convergence value of 10^{-2} were consistently worse than the rest; they were thrown out early on. For more precise stopping values, we executed training runs using 20, 40, and 50 AL iterations. However, they did not yield ideal stopping points. Specifically, even with less precise convergence, such as 10^{-3} or 10^{-4}, we can achieve results similar to, and sometimes better than, results run using more specific convergence. In fact, the results for 10^{-3} and 10^{-7} on 10 different randomly selected datasets are statistically very close (see Table 2). The average F_1-measure across all 10 models for iterations 36-40 was within a percentage point of 88% regardless of the stopping value used. Training a system on 10^{-3} instead of 10^{-7} should lead to a reduction in both computation and human effort without any significant loss of precision.

Δ*LogLi*	**Average**
10^{-3}	0.882605
10^{-4}	0.877256
10^{-5}	0.885260
10^{-6}	**0.888748**
10^{-7}	0.884872

Table 2: Average F_1-measure across 10 different *Person* models (per threshold value) for active learning iterations 36-40.

Conclusion and Future Work

This paper systematically addressed an important problem of selecting fast and robust convergence criteria in parameter optimization for maximum log-likelihood values used in CRF-based NLP tasks such as Named Entity Recognition. It showed that the relative change in log-likelihood, as opposed to the function norm and the gradient norm of the log-likelihood function, is a consistent convergence factor. The stopping threshold value of 10^{-5} for this convergence criterion is independent of both the type of the named entity and the size of the training data. In contrast to the traditional 10^{-7} stopping threshold value, using 10^{-5} achieves F_1-measures for model accuracy that are as good or better than using more precise convergence value. Reducing the threshold by two orders of magnitude significantly lessened the training time.

Using active learning with a comparable or higher F_1-measure further reduced computation time and human effort. Our active learning results highlighted the risk of over-fitting the training data for machine learning. Using a relatively high convergence value yielded F_1-measures that were approximately the same as lower convergence values that take much less time to compute. The tightest convergence values often performed worse than 10^{-5} due to over-fitting.

Our results showed that more specific convergence criteria do not necessarily produce better F_1-measure results. Further experiments may determine whether gradually tightening the convergence criteria during active learning could lead to fewer iterations of training and sentence annotation. Additionally, no studies exist that use CRFs and L-BFGS to examine convergence criteria that are appropriate for active learning cycles. Achieving a criterion for this would be an important step in analyzing over-fitting.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments that have enhanced the quality of this paper. This work is performed as part of the Scientific Data

Management Center (http://sdmcenter.lbl.gov) under the Department of Energy's Scientific Discovery through Advanced Computing program (http://www.scidac.org). The work of P. Breimyer was partially supported by the College of Engineering, North Carolina State University. Oak Ridge National Laboratory is managed by UT-Battelle for the LLC U.S. D.O.E. under contract no. DEAC05-00OR22725.

References

Brinker, K. 2003. Incorporating diversity in active learning with support vector machines. *Twentieth International Conference on Machine Learning* 20:59–66.

Dietterich, T. 1995. Overfitting and undercomputing in machine learning. *ACM Comput. Surv.* 27:326–327.

Freund, Y.; Seung, H. S.; Shamir, E.; and Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Mach. Learn.* 28:133–168.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. Morgan Kaufmann Publishers Inc.

Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Math. Program.* 45:503–528.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. *Proceedings of the 6th Conference on Natural Language Learning* 20:1–7.

McCallum, A., and Nigam, K. 1998. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 350–358. Morgan Kaufmann Publishers Inc.

Nocedal, J.; Sartenaer, A.; and Zhu, C. 2002. On the behavior of the gradient norm in the steepest descent method. *Comput. Optim. Appl.* 22:5–35.

Sarawagi, S. 2005. CRF project page. http://crf.sourceforge.net/.

Sassano, M. 2002. An empirical study of active learning with support vector machines forjapanese word segmentation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 505–512. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Shen, D.; Zhang, J.; Su, J.; Zhou, G.; and Tan, C.-L. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 589. Barcelona, Spain: Association for Computational Linguistics.

Symons, C. T.; Samatova, N. F.; Krishnamurthy, R.; Park, B. H.; Umar, T.; Buttler, D.; Critchlow, T.; and Hysom, D. 2006. Multi-criterion active learning in conditional random fields. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, 323–331. IEEE Computer Society.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* 2:45–66.

Wallach, H. 2002. *Efficient Training of Conditional Random Fields*. Ph.D. Dissertation, University of Edinburgh.

Zhu, X.; Lafferty, J.; and Ghahramani, Z. 2003. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML workshop on the Continuum from Labeled to Unlabeled Data*.