An Intrinsic Stopping Criterion for Committee-Based Active Learning

Fredrik Olsson
SICS
Box 1263
SE-164 29 Kista, Sweden
fredrik.olsson@sics.se

Katrin Tomanek
Jena University Language & Information Engineering Lab
Friedrich-Schiller-Universität Jena
Fürstengraben 30, D-07743 Jena, Germany
katrin.tomanek@uni-jena.de

Abstract

As supervised machine learning methods are increasingly used in language technology, the need for high-quality annotated language data becomes imminent. Active learning (AL) is a means to alleviate the burden of annotation. This paper addresses the problem of knowing when to stop the AL process without having the human annotator make an explicit decision on the matter. We propose and evaluate an intrinsic criterion for committee-based AL of named entity recognizers.

1 Introduction

With the increasing popularity of supervised machine learning methods in language processing, the need for high-quality labeled text becomes imminent. On the one hand, the amount of readily available texts is huge, while on the other hand the labeling and creation of corpora based on such texts is tedious, error prone and expensive.

Active learning (AL) is one way of approaching the challenge of classifier creation and data annotation. Examples of AL used in language engineering include named entity recognition (Shen et al., 2004; Tomanek et al., 2007), text categorization (Lewis and Gale, 1994; Hoi et al., 2006), part-of-speech tagging (Ringger et al., 2007), and parsing (Thompson et al., 1999; Becker and Osborne, 2005).

AL is a supervised machine learning technique in which the learner is in control of the data used for learning – the control is used to query an oracle, typically a human, for the correct label of the unlabeled training instances for which the classifier learned so far makes unreliable predictions.

The AL process takes as input a set of labeled instances and a larger set of unlabeled instances, and produces a classifier and a relatively small set of newly labeled data. The overall goal is to obtain as good a classifier as possible, without having to mark-up and supply the learner with more than necessary data. The learning process aims at keeping the human annotation effort to a minimum, only asking for advice where the training utility of the result of such a query is high.

The approaches taken to AL in this paper are based on committees of classifiers with access to pools of data. Figure 1 outlines a prototypical committee-based AL loop. In this paper we focus on the question when AL-driven annotation should be stopped (Item 7 in Figure 1).

Usually, the progress of AL is illustrated by means of a learning curve which depicts how the classifier's performance changes as a result of increasingly more labeled training data being available. A learning curve might be used to address the issue of knowing when to stop the learning process – once the curve has leveled out, that is, when additional training data does not contribute (much) to increase the performance of the classifier, the AL process may be terminated. While in a random selection scenario, classifier performance can be estimated by cross-validation on the labeled data, AL requires a held-out annotated reference corpus. In AL, the performance of the classifier cannot be reliably estimated using the data labeled in the process since sampling strategies for estimating performance assume independently and identically distributed examples (Schütze et al., 2006). The whole point in AL is to obtain a distribution of instances that is skewed in favor of the base learner used.

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pages 138–146,
Boulder, Colorado, June 2009.

1. Initialize the process by applying *EnsembleGeneration-Method* using base learner B on labeled training data set D_L to obtain a committee of classifiers C.
2. Have each classifier in C predict a label for every instance in the unlabeled data set D_U, obtain labeled set $D_U{}'$.
3. From $D_U{}'$, select the most informative n instances to learn from, obtaining $D_U{}''$.
4. Ask the teacher for classifications of the instances I in $D_U{}''$.
5. Move I, with supplied classifications, from D_U to D_L.
6. Re-train using *EnsembleGenerationMethod* and base learner B on the newly extended D_L to obtain a new committee, C.
7. Repeat steps 2 through 6 until D_U is empty or some stopping criterion is met.
8. Output classifier learned using *EnsembleGeneration-Method* and base learner B on D_L.

Figure 1: A prototypical query by committee algorithm.

In practice, however, an annotated reference corpus is rarely available and its creation would be inconsistent with the goal of creating a classifier with as little human effort as possible. Thus, other ways of deciding when to stop AL are needed. In this paper, we propose an intrinsic stopping-criterion for committee-based AL of named entity recognizers. It is intrinsic in that it relies on the characteristics of the data and the base learner[1] rather than on external parameters, i.e., the stopping criterion does not require any pre-defined thresholds.

The paper is structured as follows. Section 2 sketches interpretations of ideal stopping points and describes the idea behind our stopping criterion. Section 3 outlines related work. Section 4 describes the experiments we have conducted concerning a named entity recognition scenario, while Section 5 presents the results which are then discussed in Section 6. Section 7 concludes the paper.

2 A stopping criterion for active learning

What is the ideal stopping point for AL? Obviously, annotation should be stopped at the latest when the *best* classifier for a scenario is yielded. However, depending on the scenario at hand, the "best" classifier could have different interpretations. In many papers on AL and stopping criteria, the best (or optimal) classifier is the one that yields the highest performance on a test set. It is assumed that AL-based annotation should be stopped as soon as this performance is reached. This could be generalized as stopping criteria based on maximal classifier performance. In practice, the trade-off between annotation effort and classifier performance is related to the achievable performance given the learner configuration and data under scrutiny: For instance, would we invest many hours of additional annotation effort just to possibly increase the classifier performance by a fraction of a percent? In this context, a stopping criterion may be based on classifier performance convergence, and consequently, we can define the best possible classifier to be one which cannot learn more from the remaining pool of data.

The intrinsic stopping criterion (ISC) we propose here focuses on the latter aspect of the ideal stopping point described above – exhaustiveness of the AL pool. We suggest to stop the annotation process of the data from a given pool when the base learner cannot learn (much) more from it. The definition of our intrinsic stopping criterion for committee-based AL builds on the notions of Selection Agreement (Tomanek et al., 2007), and Validation Set Agreement (Tomanek and Hahn, 2008).

The Selection Agreement (SA) is the agreement among the members of a decision committee regarding the classification of the *most informative* instance selected from the pool of unlabeled data in each AL round. The intuition underlying the SA is that the committee will agree more on the hard instances selected from the remaining set of unlabeled data as the AL process proceeds. When the members of the committee are in complete agreement, AL *should* be aborted since it no longer contributes to the overall learning process – in this case, AL is but a computationally expensive counterpart of random sampling. However, as pointed out by Tomanek et al. (2007), the SA hardly ever signals complete agreement and can thus not be used as the sole indicator of AL having reached the point at which it should be aborted.

The Validation Set Agreement (VSA) is the agree-

[1]The term *base learner (configuration)* refers to the combination of base learner, parameter settings, and data representation.

ment among the members of the decision committee concerning the classification of a held-out, unannotated data set (the validation set). The validation set stays the same throughout the entire AL process. Thus, the VSA is mainly affected by the performance of the committee, which in turn, is grounded in the information contained in the most informative instances in the pool of unlabeled data. Tomanek and colleagues argue that the VSA is thus a good approximation of the (progression of the) learning curve and can be employed as decision support for knowing when to stop annotating – from the slope of the VSA curve one can read whether further annotation will result in increased classifier performance.

We combine the SA and the VSA into a single stopping criterion by relating the agreement of the committee on a held-out validation set with that on the (remaining) pool of unlabeled data. If the SA is larger than the VSA, it is a signal that the decision committee is more in agreement concerning the most informative instances in the (diminishing) unlabeled pool than it is concerning the validation set. This, in turn, implies that the committee would learn more from a random sample[2] from the validation set (or from a data source exhibiting the same distribution of instances), than it would from the unlabeled data pool. Based on this argument, a stopping criterion for committee-based AL can be formulated as:

> Active learning may be terminated when the Selection Agreement is larger than, or equal to, the Validation Set Agreement.

In relation to the stopping criterion based solely on SA proposed by Tomanek et al. (2007), the above defined criterion comes into effect earlier in the AL process. Furthermore, while it was claimed in (Tomanek and Hahn, 2008) that one can observe the classifier convergence from the VSA curve (as it approximated the progression of the learning curve), that requires a threshold to be specified for the actual stopping point. The ISC is completely intrinsic and does thus not require any thresholds to be set.

3 Related work

Schohn and Cohn (2000) report on document classification using AL with Support Vector Machines. If the most informative instance is no closer to the decision hyperplane than any of the support vectors, the margin has been exhausted and AL is terminated.

Vlachos (2008) suggests to use classifier confidence to define a stopping criterion for uncertainty-based sampling. The idea is to stop learning when the confidence of the classifier, on an external, possibly unannotated test set, remains at the same level or drops for a number of consecutive iterations during the AL process. Vlachos shows that the criterion indeed is applicable to the tasks he investigates.

Zhu and colleagues (Zhu and Hovy, 2007; Zhu et al., 2008a; Zhu et al., 2008b) introduce *max-confidence*, *min-error*, *minimum expected error strategy*, *overall-uncertainty*, and *classification-change* as means to terminate AL. They primarily use a single-classifier approach to word sense disambiguation and text classification in their experiments. *Max-confidence* seeks to terminate AL once the classifier is most confident in its predictions. In the *min-error* strategy, the learning is halted when there is no difference between the classifier's predictions and those labels provided by a human annotator. The *minimum expected error strategy* involves estimating the classification error on future unlabeled instances and stop the learning when the expected error is as low as possible. *Overall-uncertainty* is similar to max-confidence, but unlike the latter, overall-uncertainty takes into account all data remaining in the unlabeled pool when estimating the uncertainty of the classifier. *Classification-change* builds on the assumption that the most informative instance is the one which causes the classifier to change the predicted label of the instance. Classification-change-based stopping is realized by Zhu and colleagues such that AL is terminated once no predicted label of the instances in the unlabeled pool change during two consecutive AL iterations.

Laws and Schütze (2008) investigate three ways of terminating uncertainty-based AL for named entity recognition – *minimal absolute performance*, *maximum possible performance*, and *convergence*. The *minimal absolute performance* of the system is set by the user prior to starting the AL process. The classifier then estimates its own performance using a held-out unlabeled data set. Once the performance is reached, the learning is terminated. The *maximum possible performance* strategy refers to

[2] The sample has to be large enough to mimic the distribution of instances in the original unlabeled pool.

the optimal performance of the classifier given the data. Once the optimal performance is achieved, the process is aborted. Finally, the *convergence* criterion aims to stop the learning process when the pool of available data does not contribute to the classifier's performance. The convergence is calculated as the gradient of the classifier's estimated performance or uncertainty. Laws and Schütze conclude that both gradient-based approaches, that is, convergence, can be used as stopping criteria relative to the optimal performance achievable on a given pool of data. They also show that while their method lends itself to acceptable estimates of accuracy, it is much harder to estimate the recall of the classifier. Thus, the stopping criteria based on minimal absolute or maximum possible performance are not reliable.

The work most related to ours is that of Tomanek and colleagues (Tomanek et al., 2007; Tomanek and Hahn, 2008) who define and evaluate the *Selection Agreement* (SA) and the *Validation Set Agreement* (VSA) already introduced in Section 2. Tomanek and Hahn (2008) conclude that monitoring the progress of AL should be based on a separate validation set instead of the data directly affected by the learning process – thus, VSA is preferred over SA. Further, they find that the VSA curve approximates the progression of the learning curve and thus classifier performance convergence could be estimated. However, to actually find where to stop the annotation, a threshold needs to be set.

Our proposed intrinsic stopping criterion is unique in several ways: The ISC is intrinsic, relying only on the characteristics of the base learner and the data at hand in order to decide when the AL process may be terminated. The ISC does not require the user to set any external parameters prior to initiating the AL process. Further, the ISC is designed to work with committees of classifiers, and as such, it is independent of how the disagreement between the committee members is quantified. The ISC does neither rely on a particular base learner, nor on a particular way of creating the decision committee.

4 Experiments

To challenge the definition of the ISC, we conducted two types of experiments concerning named entity recognition. The primary focus of the first type of experiment is on creating classifiers (classifier-centric), while the second type is concerned with the creation of annotated documents (data-centric). In all experiments, the agreement among the decision committee members is quantified by the Vote Entropy measure (Engelson and Dagan, 1996):

$$VE(e) = -\frac{1}{\log k} \sum_{l} \frac{V(l,e)}{k} \log \frac{V(l,e)}{k} \qquad (1)$$

where k is the number of members in the committee, and $V(l,e)$ is the number of members assigning label l to instance e. If an instance obtains a low Vote Entropy value, it means that the committee members are in high agreement concerning its classification, and thus also that it is less a informative one.

4.1 Classifier-centric experimental settings

In common AL scenarios, the main goal of using AL is to create a good classifier with minimal label complexity. To follow this idea, we select sentences that are assumed to be useful for classifier training. We decided to select complete sentences – instead of, e.g., single tokens – as in practice annotators must see the context of words to decide on their entity labels.

Our experimental setting is based on the AL approach described by Tomanek et al. (2007): The committee consists of $k = 3$ Maximum Entropy (ME) classifiers (Berger et al., 1996). In each AL iteration, each classifier is trained on a randomly drawn (sampling without replacement) subset $L' \subset L$ with $|L'| = \frac{2}{3}L$, L being the set of all instances labeled so far (cf. *EnsembleGenerationMethod* in Figure 1). Usefulness of a sentence is estimated as the average token Vote Entropy (cf. Equation 1). In each AL iteration, the 20 most useful sentences are selected ($n = 20$ in Step 3 in Figure 1). AL is started from a randomly chosen seed of 20 sentences.

While we made use of ME classifiers during the selection, we employed an NE tagger based on Conditional Random Fields (CRF) (Lafferty et al., 2001) during evaluation time to determine the learning curves. CRFs have a significantly higher tagging performance, so the final classifier we are aiming at should be a CRF model. We have shown before (Tomanek et al., 2007) that MEs are well apt as selectors with the advantage of much shorter training times than CRFs. For both MEs and CRFs the

same features were employed which comprised orthographical (based mainly on regular expressions), lexical and morphological (suffixed/prefixed, word itself), syntactic (POS tags), as well as contextual (features of neighboring tokens) ones.

The experiments on classifier-centric AL have been performed on the English data set of corpus used in the CoNLL-2003 shared task (Tjong Kim Sang and Meulder, 2003). This corpus consists of newspaper articles annotated with respect to person, location, and organisation entities. As AL pool we took the training set which consists of about 14,000 sentences ($\approx 200,000$ tokens). As validation set and as gold standard for plotting the learning curve we used CoNLL's evaluation corpus which sums up to 3,453 sentences.

4.2 Data-centric experimental settings

While AL is commonly used to create as good classifiers as possible, with the amount of human effort kept to a minimum, it may result in fragmented and possibly non re-usable annotations (e.g., a collection of documents in which only some of the names are marked up). This experiment concerns a method of orchestrating AL in a way beneficial for the bootstrapping of annotated data (Olsson, 2008). The bootstrapping proper is realized by means of AL for selecting *documents* to annotate, as opposed to *sentences*. This way the annotated data set is comprised of entire documents thus promoting data creation. As in the classifier-centric setting, the task is to recognize names – persons, organizations, locations, times, dates, monetary expressions, and percentages – in news wire texts. The texts used are part of the MUC-7 corpus (Linguistic Data Consortium, 2001) and consists of 100 documents, 3,480 sentences, and 90,790 tokens. The task is approached using the IOB tagging scheme proposed by, e.g., Ramshaw and Marcus (1995), turning the original 7-class task into a 15-class task. Each token is represented using a fairly standard menagerie of features, including such stemming from the surface appearance of the token (e.g., *Contains dollar? Length in characters*), calculated based on linguistic pre-processing made with the English Functional Dependency Grammar (Tapanainen and Järvinen, 1997) (e.g., *Case, Part-of-speech*), fetched from precompiled lists of information (e.g., *Is first name?*), and features based on predictions concerning the context of the token (e.g, *Class of previous token*).

The decision committee is made up from 10 boosted decision trees using MultiBoostAB (Webb, 2000) (cf. *EnsembleGenerationMethod* in Figure 1). Each classifier is created by the REPTree decision tree learner described by Witten and Frank (2005). The informativeness of a document is calculated by means of average token Vote Entropy (cf. Equation 1). The seed set of the AL process consists of five randomly selected documents. In each AL iteration, one document is selected for annotation from the corpus ($n = 1$ in Step 3 in Figure 1).

5 Results

Two different scenarios were used to illustrate the applicability of the proposed intrinsic stopping criterion. In the first scenario, we assumed that the pool of unlabeled data was static and fairly large. In the second scenario, we assumed that the unlabeled data would be collected in smaller batches as it was made available on a stream, for instance, from a news feed. Both the classifier-centric and the data-centric experiments were carried out within the first scenario. Only the classifier-centric experiment was conducted in the stream-based scenario.

In the classifier-centric setting, the SA is defined as (1 − Vote Entropy) for the most informative instances in the unlabeled pool, that is, the per-token average Vote Entropy on the most informative sentences. Analogously, in the data-centric setting, the SA is defined as (1 − Vote Entropy) for the most informative document – here too, the informativeness is calculated as the per-token average Vote Entropy. In both settings, the VSA is the per-token average Vote Entropy on the validation set.

5.1 AL on static pools

The intersection of the SA and VSA agreement curves indicates a point at which the AL process may be terminated without (a significant) loss in classifier performance. For both AL scenarios (data- and classifier-centric) we plot both the learning curves for AL and random selection, as well as the SA and VSA curve for AL. In both scenarios, these

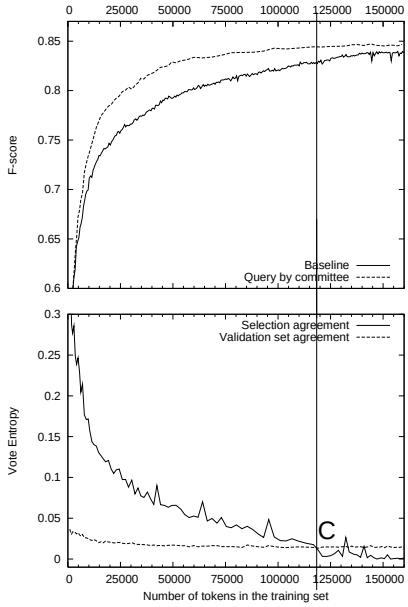

Figure 2: Classifier-centric AL experiments on the CoNLL corpus. The intersection, *C*, corresponds to the point where (almost) no further improvement in terms of classifier performance can be expected. The baseline learning curve shows the results of learning from randomly sampled data.

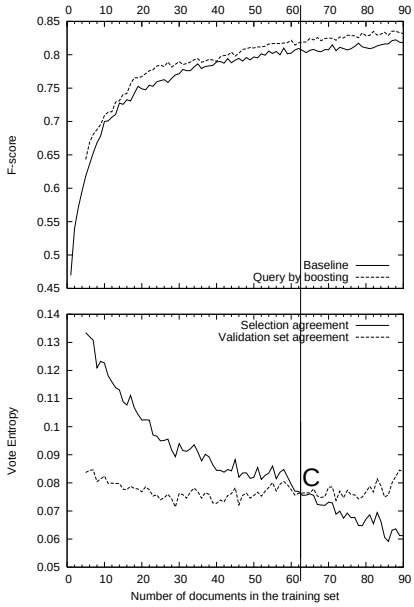

Figure 3: Data-centric AL experiments on the MUC-7 corpus. The intersection, *C*, corresponds to a point at which the AL curve has almost leveled out. The baseline learning curve shows the results of learning from randomly sampled data.

curves are averages over several runs.[3]

The results from the classifier-centric experiment on the CoNLL corpus are presented in Figure 2. AL clearly outperforms random selection. The AL curve converges at a maximum performance of $F \approx 84\%$ after about 125,000 tokens. As expected, the SA curve drops from high values in the beginning down to very low values in the end where hardly any interesting instances are left in the pool. The intersection (*C*) with the VSA curve is very close to the point (125,000 tokens) where no further increase of performance can be reached by additional annotation making it a good stopping point.

The results from the data-centric experiment are available in Figure 3. The bottom part shows the SA and VSA curves. The ISC occurs at the intersection of the SA and VSA curves (*C*), which corresponds to a point well beyond the steepest part of the learning curve. While stopping the learning at *C* results in a classifier with performance inferior what is maximally achievable, stopping at *C* arguably corresponds to a plausible place to abort the learning. The optimal performance is $F \approx 83.5\%$, while the ISC corresponds to $F \approx 82\%$.

Keep in mind that the learning curves with which the ISC are compared are not available in a practical situation, they are included in Figures 2 and 3 for the sake of clarity only.

5.2 AL on streamed data

One way of paraphrasing the ISC is: Once the intersection between the SA and VSA curves has been reached, the most informative instances remaining in the pool of unlabeled data are less informative to the classifier than the instances in the held-out, unlabeled validation set are on average. This means that the classifier would learn more from a sufficiently large sample taken from the validation set than it would if the AL process continued on the remaining unlabeled pool.[4]

As an illustration of the practical applicability of the ISC consider the following scenario. Assume

[3]The classifier-centric experiments are averages over three independent runs. The data-centric experiments are averages over ten independent runs.

[4]Note however, that the classifier might still learn from the instances in the unlabeled pool – applying the ISC only means that the classifier would learn more from a validation set-like distribution of instances.

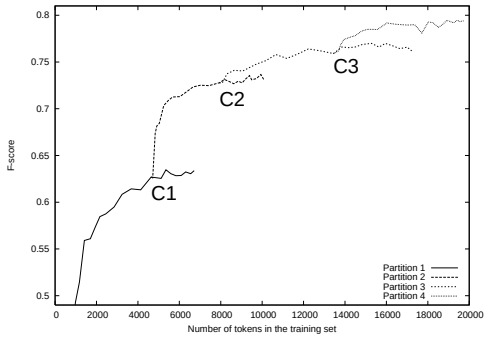

Figure 4: AL curves for the four partitions used in the experiments on streamed data. C_i denotes a point at which the AL is terminated for partition i and a new partition is employed instead. *C1* corresponds to the ISC plotted in the graph labeled *Partition 1* in Figure 5, *C2* to the ISC in *Partition 2*, and *C3* to the ISC in *Partition 3*.

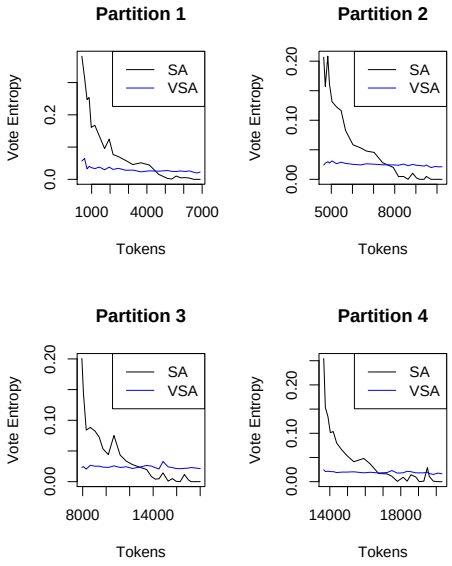

Figure 5: The SA and VSA curves for the four data partitions used in the experiment on streamed data. Each intersection – ISC – corresponds to a point where AL is terminated.

that we are collecting data from a stream, for instance items taken from a news feed. Thus, the data is not available on the form of a closed set, but rather an open one which grows over time. To make the most of the human annotators in this scenario, we want them to operate on batches of data instead of annotating individual news items as they are published. The purpose of the annotation is to mark up names in the texts in order to train a named entity recognizer. To do so, we wait until there has appeared a given number of sentences on the stream, and then collect those sentences. The problem is, how do we know when the AL-based annotation process for each such batch should be terminated? We clearly do not want the annotators to annotate all sentences, and we cannot have the annotators set new thresholds pertaining to the absolute performance of the named entity recognizer for each new batch of data available. By using the ISC, we are able to automatically issue a halting of the AL process (and thus also the annotation process) and proceed to the next batch of data without losing too much in performance, and without having the annotators mark up too much of the available data. To this end, the ISC seems like a reasonable trade-off between annotation effort and performance gain.

To carry out this experiment we took a sub sample of 10% (1,400 sentences) from the original AL pool of the CoNLL corpus as validation set.[5] The rest of this pool was split into batches of about 500 consecutive sentences. Classifier-centric AL was now run taking the first batch as pool to select from. At the point where the SA and VSA curve crossed, we continued AL selection from the next batch and so forth. Figure 4 shows the learning curve for a simulation of the scenario described above. The intersection between the SA and VSA curves for partition 1 as depicted in Figure 5 corresponds to the first "step" (ending in *C1*) in the stair-like learning curve in Figure 4. The step occurs after 4,641 tokens. Analogously, the other steps (ending in *C2* and *C3*, respectively) in the learning curve corresponds the intersection between the SA and VSA curves for partitions 2 and 3 in Figure 5. The intersection for partition 4 corresponds to the point were we would have turned to the next partition. This experiment was stopped after 4 partitions.

Table 1 shows the accumulated number of sentences and tokens (center columns) that required annotation in order to reach the ISC for each partition. In addition, the last column in the table shows the number of sentences (of the 500 collected for inclu-

[5]Note that the original CoNLL test set was not used in this experiment, thus the F-score reported in Figure 4 cannot be compared to that in Figure 2.

Partition	Sents	Toks	Sentences per partition
1	320	4,641	320
2	580	7,932	260
3	840	13,444	260
4	1070	16,751	230

Table 1: The number of tokens and sentences required to reach the ISC for each partition.

sion in each partition) needed to reach the ISC – each new partition contributes less to the increase in performance than the preceding ones.

6 Discussion

We have argued that one interpretation of the ISC is that it constitutes the point where the informativeness on the remaining part of the AL pool is lower than the informativeness on a different and independent data set with the same distribution. In the first AL scenario where there is one static pool to select from, reaching this point can be interpreted as an overall stopping point for annotation. Here, the ISC represents a trade-off between the amount of data annotated and the classifier performance obtained such that the resulting classifier is nearly optimal with respect to the data at hand. In the second, stream-based AL scenario where several smaller partitions are consecutively made available to the learner, the ISC serves as an indicator that the annotation of one batch should be terminated, and that the mark-up should proceed with the next batch.

The ISC constitutes an intrinsic way of determining when to stop the learning process. It does not require any external parameters such as pre-defined thresholds to be set, and it depends only on the characteristics of the data and base learner at hand. The ISC can be utilized to relate the performance of the classifier to the performance that is possible to obtain by the data and learner at hand.

The ISC can not be used to estimate the performance of the classifier. Consequently, it can not be used to relate the classifier's performance to an externally set level, such as a particular F-score provided by the user. In this sense, the ISC may serve as a complement to stopping criteria requiring the classifier to achieve absolute performance measures before the learning process is aborted, for instance the *max-confidence* proposed by Zhu and Hovy (2007), and the *minimal absolute performance* introduced by Laws and Schütze (2008).

7 Conclusions and Future Work

We have defined and empirically tested an intrinsic stopping criterion (ISC) for committee-based AL. The results of our experiments in two named entity recognition scenarios show that the stopping criterion is indeed a viable one, which represents a fair trade-off between data use and classifier performance. In a setting in which the unlabeled pool of data used for learning is static, terminating the learning process by means of the ISC results in a nearly optimal classifier. The ISC can also be used for deciding when the pool of unlabeled data needs to be refreshed.

We have focused on challenging the ISC with respect to named entity recognition, approached in two very different settings; future work includes experiments using the ISC for other tasks. We believe that the ISC is likely to work in AL-based approaches to, e.g., part-of-speech tagging, and chunking as well. It should be kept in mind that while the types of experiments conducted here concern the same task, the ways they are realized differ in many respects: the ways the decision committees are formed, the data sets used, the representation of instances, the relation between the sample size and the instance size, as well as the pre-processing tools used. Despite these differences, which outnumbers the similarities, the ISC proves a viable stopping criterion.

An assumption underlying the ISC is that the initial distribution of instances in the pool of unlabeled data used for learning, and the distribution of instances in the validation set are the same (or at least very similar). Future work also includes investigations of automatic ways to ensure that this assumption is met.

Acknowledgements

The first author was funded by the EC project COMPANIONS (IST-FP6-034434), the second author was funded by the EC projects BOOTStrep (FP6-028099) and CALBC (FP7-231727).

References

Markus Becker and Miles Osborne. 2005. A Two-Stage Method for Active Learning of Statistical Grammars. In *Proc 19th IJCAI*, Edinburgh, Scotland, UK.

Adam Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Sean P. Engelson and Ido Dagan. 1996. Minimizing Manual Annotation Cost In Supervised Training From Corpora. In *Proc 34th ACL*, Santa Cruz, California, USA.

Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. 2006. Large-Scale Text Categorization by Batch Mode Active Learning. In *Proc 15th WWW*, Edinburgh, Scotland.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc 18th ICML*, Williamstown, Massachusetts, USA.

Florian Laws and Hinrich Schütze. 2008. Stopping Criteria for Active Learning of Named Entity Recognition. In *Proc 22nd COLING*, Manchester, England.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proc 17th ACM-SIGIR*, Dublin, Ireland.

Linguistic Data Consortium. 2001. Message understanding conference (muc) 7. LDC2001T02. FTP FILE. Philadelphia: Linguistic Data Consortium.

Fredrik Olsson. 2008. *Bootstrapping Named Entity Annotation by means of Active Machine Learning – A Method for Creating Corpora*. Ph.D. thesis, Department of Swedish, University of Gothenburg.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text Chunking using Transformation Based Learning. In *Proc 3rd VLC*, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

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 *Proc Linguistic Annotation Workshop*, Prague, Czech Republic.

Greg Schohn and David Cohn. 2000. Less is More: Active Learning with Support Vector Machines. In *Proc 17th ICML*, Stanford University, Stanford, California, USA.

Hinrich Schütze, Emre Velipasaoglu, and Jan O. Pedersen. 2006. Performance Thresholding in Practical Text Classification. In *Proc 15th CIKM*, Arlington, Virginia, USA.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-Criteria-based Active Learning for Named Entity Recognition. In *Proc 42nd ACL*, Barcelona, Spain.

Pasi Tapanainen and Timo Järvinen. 1997. A Non-Projective Dependency Parser. In *Proc 5th ANLP*, Washington DC, USA.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active Learning for Natural Language Parsing and Information Extraction. In *Proc 16th ICML*, Bled, Slovenia.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language Independent Named Entity Recognition. In *Proc 7th CoNLL*, Edmonton, Alberta, Canada.

Katrin Tomanek and Udo Hahn. 2008. Approximating Learning Curves for Active-Learning-Driven Annotation. In *Proc 6th LREC*, Marrakech, Morocco.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. An Approach to Text Corpus Construction which Cuts Annotation Costs and Maintains Reusability of Annotated Data. In *Proc Joint EMNLP-CoNLL*, Prague, Czech Republic.

Andreas Vlachos. 2008. A Stopping Criterion for Active Learning. *Computer, Speech and Language*, 22(3):295–312, July.

Geoffrey I. Webb. 2000. MultiBoosting: A Technique for Combining Boosting and Wagging. *Machine Learning*, 40(2):159–196, August.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools with Java Implementations. 2nd Edition.* Morgan Kaufmann, San Fransisco.

Jingbo Zhu and Eduard Hovy. 2007. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. In *Proc Joint EMNLP-CoNLL*, Prague, Czech Republic.

Jingbo Zhu, Huizhen Wang, and Eduard Hovy. 2008a. Learning a Stopping Criterion for Active Learning for Word Sense Disambiguation and Text Classification. In *Proc 3rd IJCNLP*, Hyderabad, India.

Jingbo Zhu, Huizhen Wang, and Eduard Hovy. 2008b. Multi-Criteria-based Strategy to Stop Active Learning for Data Annotation. In *Proc 22nd COLING*, Manchester, England.