A stopping criterion for active learning

Andreas Vlachos

15 JJ Thomson Avenue, Cambridge, UK

Abstract

Active learning (AL) is a framework that attempts to reduce the cost of annotating training material for statistical learning methods. While a lot of papers have been presented on applying AL to natural language processing tasks reporting impressive savings, little work has been done on defining a stopping criterion. In this work, we present a stopping criterion for active learning based on the way instances are selected during uncertainty-based sampling and verify its applicability in a variety of settings. The statistical learning models used in our study are support vector machines (SVMs), maximum entropy models and Bayesian logistic regression and the tasks performed are text classification, named entity recognition and shallow parsing. In addition, we present a method for multiclass mutually exclusive SVM active learning.

Key words: active learning, SVMs, text classification, NER

1 Introduction

One of the most important issues when applying statistical learning techniques is the creation of training material. However, manual annotation involves a substantial amount of human effort and becomes the main bottleneck when attempting new tasks or porting existing techniques to new domains. For example, the recent interest in the biomedical domain required the creation of annotated datasets such as GENIA (Kim et al., 2003) in order to allow existing methods to be evaluated and new ones to be developed.

In order to overcome the lack of annotated material, methods were developed that either take advantage of extant resources (Morgan et al., 2004) to create training material automatically, or use some seed patterns and iteratively

Email address: av308@cl.cam.ac.uk (Andreas Vlachos).

bootstrap a classifier (Agichtein and Gravano, 2000). However, such methods have limitations in their performance and their applicability.

A different approach to this issue is the use of active learning (AL) (Cohn et al., 1994; Seung et al., 1992). In this framework, the supervised classifier selects the instances that are likely to be the most informative to train on. Then the selected instances are annotated by a human, added to the training material of the classifier and the loop is repeated. This is in contrast with the traditional method of annotating randomly selected material, referred to as random sampling. Active learning has been used successfully in many tasks with a variety of classifiers. Examples include text classification using support vector machines (Tong and Koller, 2001) and parse selection using log-linear models (Baldridge and Osborne, 2007). In most cases, the savings in annotated instances compared to random selection used were substantial.

An issue that has not been studied extensively is the definition of an active learning stopping criterion. In most cases, the experimental setup consists of a set of candidate training instances and a test set used to evaluate the performance achieved. The savings achieved are commonly reported as the percentage of the candidate training instances selected during active learning to achieve the same performance as when using all the available training instances, or as the difference in performance between active learning and random selection for a given amount of annotation. This however implies that all the candidate training instances are annotated in advance. In a more realistic setting, active learning would stop when the desired performance level has been reached (Li and Sethi, 2006). Still, at least a test set has to be annotated in advance in order to measure the performance, which is a difficult and expensive process involving human effort and is exactly what we aim to minimize with active learning. Ideally, we would like to have a way of terminating the active learning process without having to use a pre-annotated dataset.

This paper presents a stopping criterion for active learning based on observations on the nature of the selections made and the behaviour of statistical learning methods. Section 2 contains a brief introduction to active learning. Section 3 describes the stopping criterion suggested and Section 4 presents experiments that demonstrate its applicability in a variety of tasks and statistical learning methods. In Section 5 we introduce a method to perform mutually exclusive multiclass SVM active learning and we present results on named entity recognition and shallow parsing. Finally, Section 6 discusses our results and related work and Section 7 concludes suggesting future work.

2 Active learning

In the active learning framework, the statistical learning model iteratively selects the instances on which it is going to be trained on. In the widely used pool-based approach[1], we start with a small labelled training set L and a large pool of unlabelled data U. In each round, a model is trained on L and it is used in order to select a batch b of instances from U which are considered to be informative. These are annotated by a human, added to L and the loop is repeated.

The main point of differentiation among the various active learning algorithms is the method of assessing the informativity of an instance. The two most popular active learning methods used in NLP are uncertainty-based sampling (Cohn et al., 1994) and query by committee (Seung et al., 1992). In uncertainty-based learning, the instances selected to be annotated are those on which the classifier is least certain of their classification. The assumption is that instances which are harder to classify are more useful to train the classifier on. The uncertainty of the classifier is commonly estimated using the entropy of its output in the case of probabilistic models. For non-probabilistic ones, the classification margin is used, as in the case of support vector machines (Tong and Koller, 2001; Schohn and Cohn, 2000; Campbell et al., 2000). The algorithm for uncertainty-based sampling appears in Figure 1:

```
Input:
seed labelled data $L$, unlabelled data $U$,
batch size $b$
Initialization:
Train a model on $L$
Active Learning Loop:
Until a stopping criterion is satisfied:
    Apply the trained model classifier on $U$
    Rank the instances in $U$ using the uncertainty of the model
    Annotate the top $b$ instances and add them to $L$
    Train the model on the expanded $L$
```

Fig. 1. Active Learning using uncertainty-based sampling

In query by committee, a committee of classifiers is trained on L, then applied to the instances of U and those which result in the highest disagreement among the classifiers are considered to be the most informative. Common ways of estimating the disagreement are the vote-entropy (Argamon-Engelson and Dagan, 1999) and the Kullback-Leibler divergence (Pereira et al., 1993;

[1] Baram et al. (2004) contains a detailed overview of various active learning approaches.

McCallum and Nigam, 1998; Becker and Osborne, 2005). In this work, we will concentrate on uncertainty-based sampling.

3 Towards a stopping criterion

An obvious stopping criterion for active learning would be to measure the performance of the trained classifier on an annotated dataset and terminate the procedure when the performance ceases to improve or when it improves at a non-satisfactory rate. However, this might not be ideal. Apart from the costs involved in creating a test set, there is also the risk that it might not be representative of what can be learnt from the pool of unlabelled data. This is likely to occur because the annotated dataset is probably going to be much smaller than the pool due to its creation cost. Therefore, using the performance on an annotated dataset as a stopping criterion could be misleading, since it is possible that informative instances from the pool would not affect the performance on that particular dataset.

Ideally, we would like to terminate active learning when there are no informative instances left in the pool, unless the budget for annotation is exhausted first. Since measuring the performance on an annotated dataset is expensive and not necessarily appropriate, we focused on using the confidence of the classifier. The latter is expected to increase as we add more instances to the training data, since the classifier obtains more information about the task. However, adding instances to the training data that contradict the information gathered already by the model, would cause the confidence to drop, since existing features become weaker cues for classification. For example, assume that we want to classify documents according to whether they are related to finance or not. The token “bank” is likely to be an indicative feature to such documents, however it can be found in non-financial documents as well with its alternative sense (as in the “river bank”). Adding non-financial documents that contain this token to the training data of a classifier is likely to reduce the strength of the feature, while there are probably more indicative features for non-financial documents, such as “river” that would cover such cases.

During uncertainty-based sampling, the instances added are those on which the classifier is most uncertain. The classifier is expected to be uncertain on instances that are dissimilar to the ones that it has been trained on. These are unlikely to contradict the knowledge gathered already by the classifier, because this would mean that the classifier would have been able to make a confident (not necessarily correct) prediction. Progressively, the classifier accumulates a larger training set to learn from and its confidence increases with its performance. Eventually, the instances left unlabelled in the pool are such that the classifier is confident of their label because they are covered by the ones already added to the training set. At this point, the sampling process can only select instances that do not contribute novel information to the classifier because they are similar to the ones already included in its training set. Interestingly, if the labels obtained for those instances contradict the predictions of the classifier then they become evidence against the information gathered in earlier rounds. As a result, the confidence of the classifier either remains at the same level, or it drops in the case that contradictory instances are encountered.

A point that needs further explanation is how instances from the pool can contradict what has been learnt already, assuming that we are dealing with a well-defined task and that they are not annotation inconsistencies. When performing a natural language processing task, we represent the instances with a feature set that is limited compared to the human background knowledge about the task. For example, the standard "bag of words" document representation used in text classification ignores word order information. Moreover, the statistical models themselves have their own limitations which do not allow them to "explain" the differences between instances. For example, linear kernel SVMs assume that the classes in the data are linearly separable. For these reasons, it is very likely that a statistical model given a feature representation cannot "explain" all the instances in a dataset, which therefore appear contradictory and reduce its confidence. As a consequence, we expect the confidence of the classifier to exhibit a rise-peak-drop pattern during uncertainty-based sampling.

In order to estimate the confidence of the classifier we apply it on a separate dataset. The latter does not need to be annotated, only feature extraction needs to be performed. Estimating the confidence of the classifier on a particular dataset involves the same risk as evaluating the performance on it, since it might not be representative, therefore it could be misleading. However, unlike performance evaluation, we don't need the dataset to be annotated and since the feature extraction is automatic we can obtain a big dataset which would minimize this risk.

In practice, after each round of uncertainty-based sampling we run the classifier over a separate, large dataset and estimate its confidence. When the confidence of the classifier drops, this suggests that the statistical model - given the feature representation used- cannot take advantage of the remaining instances in the pool of unlabelled data. It must be stressed here that while further annotation from the current pool would not benefit the model in question, a new pool could contain useful instances. Also, even if the model used cannot take advantage of more instances from the current pool, a stronger model with a better feature representation possibly could. Therefore, the remaining instances should be considered redundant only for the given model and feature representation. Further support for this is provided by the work of Baldridge and Osborne (2004), who found in their experiments that reusing material selected during active learning with a different classifier and/or feature representation is not very effective and can yield worse results than random selection.

4 Experiments

In this section we present active learning experiments in which we examine the applicability of the stopping criterion suggested in the previous section. In the subsections that follow, we discuss the statistical learning models used and the text classification and named entity recognition experiments performed.

4.1 Uncertainty-based sampling using Support Vector Machines

Support vector machines (SVMs) (Vapnik, 1995) are a state-of-the-art statistical learning model. They have been used successfully in a variety of tasks, including text classification (Joachims, 1998b) and handwritten digit recognition (LeCun et al., 1995). A training dataset D comprising of two classes $\{-1, +1\}$ is projected to a (possibly) higher dimensional space and a maximum margin separating hyperplane is found between the two classes. The separating hyperplane is defined by a set of datapoints $\{x_1, ..., x_n\}$ and their labels $\{y_1, ..., y_n\}$ which are the support vectors. Each of these datapoints is assigned a weight $a_1, ...a_n$. The projection to the higher dimensional space is performed using a suitable kernel function $K(x_i, x_j)$, which allows the calculations to take place in the original dimensional space. During classification, the test datapoints are classified according to the side of the separating hyperplane on which they are found to lie. For a datapoint x, this is performed using the following function:

$$f(x, a) = sign(\sum_{i=1}^{n} y_i a_i K(x, x_i) + b) \tag{1}$$

The sign of the weighted sum of the inner products of the datapoint with the support vectors denotes the class. Its absolute value is the distance (margin) of the datapoint from the separating boundary. This should not be confused with the probability estimates that can be obtained from other statistical learning models. It ranges from 0 to ∞ and most importantly, the margins yielded by different SVM models are not comparable with each other because different datasets and/or kernel functions define different spaces.

The choice of the kernel function defines the space in which the data is projected and it is very important because it affects the shape of the separating hyperplane to be discovered. For example, the simple and widely used linear kernel function, $K(x_i, x_j) = x_i \cdot x_j$, can only define linear separating hyperplanes. However, non-linear kernel functions such as the Gaussian ($K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$) can discover more complex hyperplanes, which in turn can result in better classification performance (Burges, 1998). The superior performance of the Gaussian kernel in particular has been verified by various authors (Joachims, 1998b; Keerthi and Lin, 2003). However, they are slower to train and parameter optimization is required (normally performed through cross-validation on the training data) in order to obtain good performance, which are the main reasons why they have not been adopted more widely in NLP. On the contrary, linear kernel SVMs can be implemented to run much faster and the parameters can be set quite efficiently without cross-validation, as in SVM-Light (Joachims, 1998a).

The standard way of performing uncertainty-based sampling is by using the entropy of the class distribution of the classifier over each instance as a measure of uncertainty. However, as mentioned earlier, SVMs do not yield probabilistic output, but a decision margin (Eq. 1). Tong and Koller (2001), Schohn and Cohn (2000) and Campbell et al. (2000) independently presented a way of performing uncertainty-based sampling with SVMs. They used the decision margin of the classifier as an indication of its uncertainty in classifying a particular instance. The assumption is that the closer a datapoint lies to the separating hyperplane, the more informative it is going to be. In each round, the instances of the pool of unlabelled data are ranked according to the margin yielded by the SVM classifier and a batch of the top-ranked instances is selected for annotation.

It is worth noting that the probabilistic outputs that can be obtained by fitting a sigmoid function (Platt, 1999) would not change the ranking of the instances in the pool of unlabelled data, since the probability estimate of an instance being positive increases monotonically with the decision margin. Tong and Koller (2001) presented an extension to the selection method described in this section, which while it is more efficient, requires training an SVM classifier twice for each instance in the unlabelled pool and therefore can be very expensive.

4.2 Text classification experiments

In the experiments of this subsection we used the Reuters RCV1-v2 corpus (Lewis et al., 2004) to perform text classification (TC). We used the files provided in the on-line appendix 12 which contain 804,414 documents, tokenized, stemmed and with the stopwords removed. We sorted them according to their publication date (older to newer) and we kept the oldest 20% of them as our training pool in our experiments. The first 100 documents of the training pool were used as seed labelled data in order to initiate the active learning process. The remaining 80% of the documents were used as our test set. We think that since the dates of the documents are available, a chronological split is a more realistic setup than a random split.

We used the SVMlight implementation of SVMs (Joachims, 1998a) in order to build a binary classifier for the most popular topic of the corpus *CCAT*, which contains 381,327 documents. Following Lewis et al. (2004), we performed feature weighting using Cornell ltc (Buckley et al., 1994), which is a variant of TF-IDF weighting. Two active learning runs were performed, with 1% (1607 documents) and 0.1% (160 documents) of the training pool added to the training data in each round. The results using the linear kernel with default parameters are presented in Figure 2.

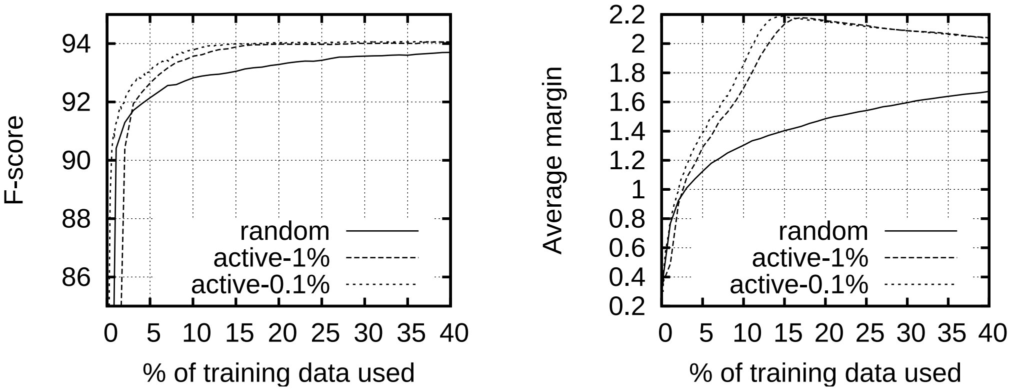

Fig. 2. Graphs for the performance (left) and the confidence (right) of linear kernel SVMs for the class *CCAT* of RCV1.

The left graph presents a typical comparison of the performance curves during active learning versus random selection. The performance during AL rises faster, especially when fewer instances are selected in each round. The performance during random sampling increases at a slower rate, reaching 94% F-score when 90% of the data are used, as opposed to 29% during AL with 1% of the data added in each round and 17% with 0.1% of the data is added. The performance of the classifier using all the available data is 94.05%.

The right graph presents the confidence curves for these runs. The confidence of the SVM classifier was estimated as the sum of the decision margins for the instances of the test set. As expected, during active learning the confidence curves exhibit a rise-peak-drop pattern. The peak coincided with the flattening of the performance curve (at about 20% of the data being used), confirming the applicability of the criterion. While it might have been more appropriate to terminate active learning earlier than that since the performance gains after

10% are limited, one should keep in mind that, as suggested in Section 3 the test dataset might not necessarily contain instances that could test what can be learnt from the pool. Reducing the batch size leads to faster fulfillment of the stopping criterion since the classifier can assess the informativity of the unlabelled instances more frequently. During random selection the confidence curve follows the performance curve, which is expected since instances contradictory to the knowledge gathered already by the model are added at a steady rate.

It is also worth noting that the stopping criterion is satisfied before the maximum performance is reached. However, the performance loss is rather small compared to the savings in annotation. In the experiments using linear kernel SVMs (Figure 2), assuming that we would terminate active learning when 20% of the data has been used (this is when a consistent drop in performance is observed), the performance achieved is 93.98%, which is lower than the maximum performance achieved (94.08%). However, to achieve the latter, 51% of the data has to be annotated, i.e. 31% of the data has to be annotated in order to gain 0.08% in F-score. Refreshing the pool of unlabelled data with new and potentially more informative instances is likely to yield higher gains.

Using the same experimental setup, we ran experiments using the Gaussian kernel provided in SVM-Light. As mentioned in Section 4.1, the Gaussian kernel requires optimization of its parameters in order to yield good results, which is usually performed using cross-validation on the training set. Ideally, in our active learning experiments the parameters should be optimized in each round using the respective training data. However, this would have been prohibitively expensive. Therefore, as a compromise, we performed cross-validation using a sample of 5,000 randomly selected documents from the training pool (approximately 3% of the total available) for parameter estimation using the grid search procedure provided in the LIBSVM toolkit (Chang and Lin, 2001). We used the parameters found throughout all our experiments with the Gaussian kernel. We performed active learning adding 1% of the unlabelled pool in each round, as well as random selection for comparison.

The curves in the left graph of Figure 3 demonstrate the efficiency of uncertainty-based sampling using Gaussian kernel SVMs. During active learning with Gaussian kernel SVMs, 94% F-score is achieved using 11% of the data (compared to 29% with linear kernel SVMs), while during random selection it is achieved using 31% of the data. Near maximum performance (94.5% F-score) is achieved using 20% and 78% of the data respectively. For comparison, using linear kernel SVMs active learning reached 94% F-score using 29% of the available data.

It must be noted that the performance of Gaussian kernel SVMs in the initial rounds (until 5% of the data has been used) is better using random selection

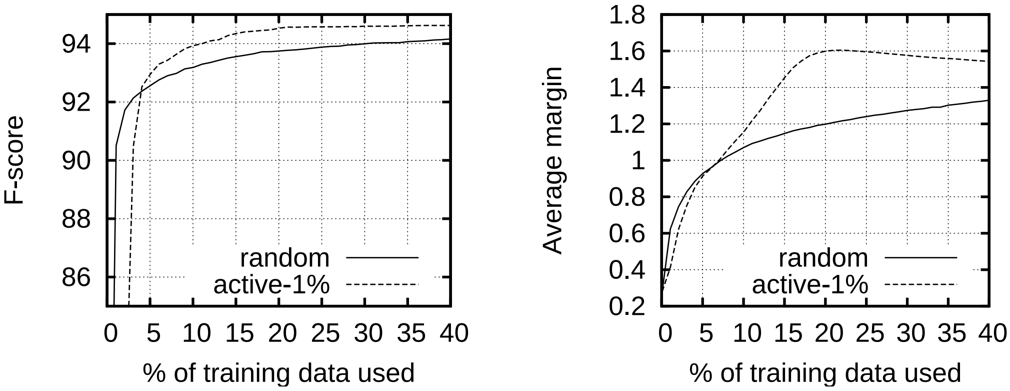

Fig. 3. Graphs for the performance (left) and the confidence (right) for Gaussian kernel SVMs for the class *CCAT* of RCV1.

than adding 1% of the data using uncertainty-based sampling. Campbell et al. (2000) suggested that random selection can be better than uncertainty-based sampling in the initial stages because the selecting model is not trained on enough data. Given that our seed data is small and from a restricted time period it is unlikely to provide good coverage of the data, so this is very likely to be the case in this experiment. In order to verify this claim, we reduced the number of instances selected in each round to 0.1% of the pool and the effect of non-optimal initial selections was alleviated. After selecting 1% of the pool in batches of 0.1% the F-score achieved was 91%, compared to 90.5% during random sampling. It is worth mentioning that Gaussian kernel SVM active learning adding 0.1% of the pool in each round achieved 94% F-score using only 8.2% of the data.

Concerning the confidence curves in the right graph of Figure 3, we observed the same rise-peak-drop pattern that was exhibited in the linear kernel SVM experiments (right graph of Figure 2). Of interest is the observation that during AL, the confidence of the Gaussian SVM model starts dropping later than that of the linear kernel (25% compared to 20%), confirming the expectation that since it can identify more complex boundaries it can take advantage of more data.

In order to verify that the cause of the drop in the confidence is the addition to the training data of instances contradicting the knowledge already gathered by the classifier, we tried to estimate the amount of contradictory information added in each round. We considered that contradictory information is added each time an instance whose label was predicted incorrectly by the classifier is selected to be added to the training data. An indication of the amount of the contradictory information added by each instance is considered to be the margin by which it is classified by the classifier divided by the average margin of the classifier in that round. This was deemed necessary as the confidence of the classifier changes in each round, resulting in the margins themselves not being comparable since the more confident the classifier becomes, the larger the margins on the incorrect predictions will be, thus concealing the effect of the actual contradictory information added by the instances themselves. To estimate the total amount of contradictory information added in each round, we summed the contradictory information contributed by all the incorrectly predicted instances selected in each round. This can be summarized in the following equation:

$$Contradictory\_information(t) = \sum_{i \in i^t} \frac{| f^t(x_i) |}{\overline{| f^t(x) |}} \qquad (2)$$

Where t is the round, i^t is the list of instances selected for annotation in round t whose label was predicted incorrectly by the classifier, $f^t(x_i)$ is the decision margin of the classifier that round for instance x_i and $\overline{| f^t(x) |}$ is the average decsion margin of the classifier in that round.

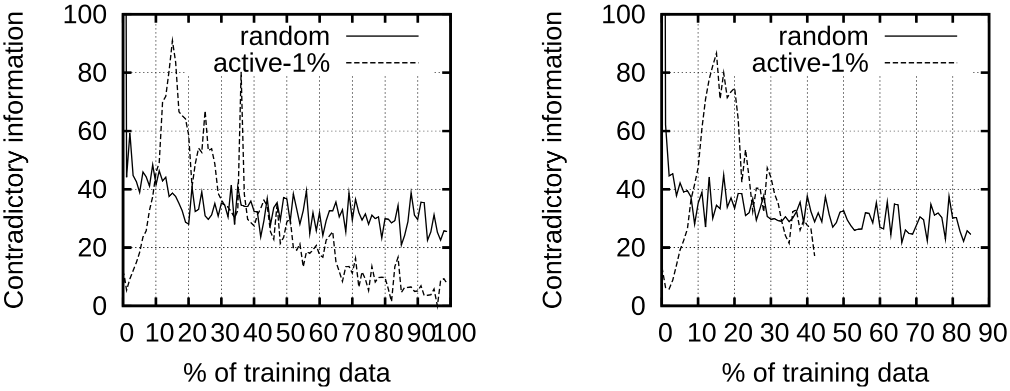

Fig. 4. Measuring the amount of contradictory information added in each round for the linear (left) and the Gaussian (right) kernel.

In Figure 4 we calculated in each round the sum of the margins on the instances chosen for annotation whose label was predicted incorrectly by the SVM classifier, divided by the average margin of the classifier over the instances of the test set. The higher this is, the more information contradictory to what has already been learnt is being added in that round. The graphs present curves for active learning and random selection using linear (left) and Gaussian (right) kernel. In both cases, it is observed that during random selection, the amount of contradictory information added in each round is roughly the same, suggesting that the contradictory information is evenly distributed throughout the experiment, which is expected since there is no bias in the selection. In the case of active learning though, the contradictory information added in each round is very little initially and it rises in the following rounds, causing the drop in the confidence. In later rounds, the amount of contradictory information added in each round drops substantially. This confirms the prediction of Section 3, that during AL the classifier initially avoids contradictory instances by selecting those on which it is most uncertain, but later on it is forced to select them and add them to its training data, thus reducing its confidence.

We tried to see if this result was related to the number of support vectors. However, in all our experiments the number of datapoints identified as support vectors during training increases in each round until it reaches a plateau, albeit more quickly during active learning compared to random selection.

4.3 Effect of class distribution

In the text classification experiments of Section 4.2, following the experimental setup of Tong and Koller (2001) and Schohn and Cohn (2000), who used the most popular classes in the Reuters-21578 dataset and obtained similar results for all of them, we used the most popular class of the RCV1 dataset. In order to investigate how SVM active learning behaves (and the applicability of the stopping criterion suggested) in the case of very imbalanced tasks, we applied it using the linear kernel to one of the least frequent classes in the dataset, *C16*, which contains 1,920 documents (0.2% of the total documents). The results presented in Figure 5 show that AL is much more efficient than random selection, which is essentialy due to the fact that AL selects almost all of the (few) positive instances in the initial rounds. It must be noted however that the maximum performance reached is 50% in F-score, far lower than the performances achieved in the experiments for the class *CCAT*. The rise-peak-drop pattern in the confidence is not exhibited as distinctly as in the previous experiments. As explained in Section 3, the satisfaction of the stopping criterion depends on the existence of instances in the pool that contradict the ones already added to the training data. If there are very few instances from one class, then it becomes less likely to find contradictory instances because the data for that class is very sparse and most of its instances are selected in the initial rounds. As a result, most instances can be added to the training data without contradicting the model built already.

4.4 Experiments with other classifiers

In order to verify the applicability of the stopping criterion with statistical learning models other than SVMs, we performed binary text classification experiments with Bayesian logistic regression (BLR) (Genkin et al., 2006) and maximum entropy (Berger et al., 1996). Logistic regression models are of

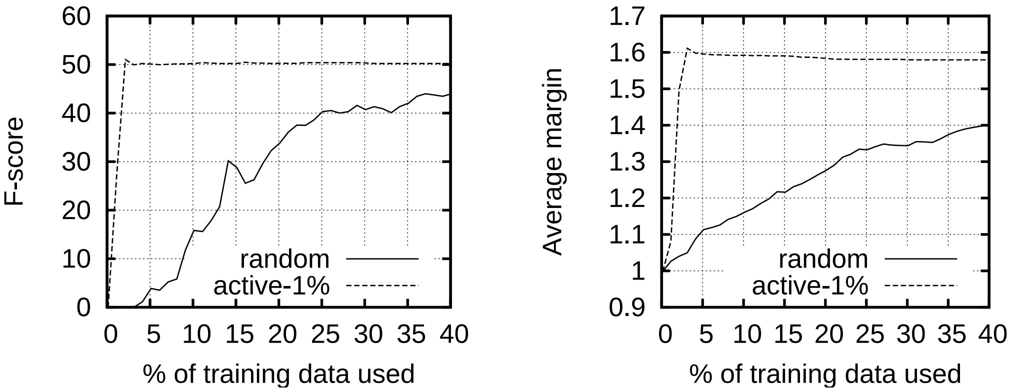

Fig. 5. Graphs for the performance (left) and the confidence (right) for linear kernel SVMs for class *C16* of RCV1.

the form:

$$p(y = +1|\beta, x) = \frac{exp(x\beta^T)}{1 + exp(x\beta^T)} \tag{3}$$

Where $y \in \{+1, -1\}$ is a binary class label, x is the feature vector representation of the instance to be classified and β is the feature weight vector which is learnt from the training data. In Bayesian logistic regression, a prior distribution on β is used. In order to perform uncertainty-based sampling using Bayesian logistic regression we followed Schein and Ungar (2004) who used the entropy of the class distribution predicted by the classifier for each instance[2]. Maximum entropy models are of the form:

$$p(y|x) = \frac{1}{Z(x)} exp(\sum_{i=1}^{k} \lambda_i f_i(x, y)) \tag{4}$$

Where y is the class label (does not need to be binary), x is the feature representation of the instance to be classified, $Z(x)$ a normalization factor, $f_i(x, y)$ are binary functions over the instance representing the features and λ_i are their respective weights. As in the case of BLR, uncertainty-based sampling was performed by selecting the instances with the highest entropy over the class distribution predicted by the classifier.

The software used for the Bayesian logistic regression experiments was BBR[3]. We optimized the parameters using the *–autosearch* option provided using 3000 randomly selected instances. For the maximum entropy experiments, we

[2] In the same work, the authors also suggest that uncertainty-based sampling is inferior to other active learning approaches when used with BLR. However, this is not the focus of this work.

[3] http://www.stat.rutgers.edu/ madigan/BBR/

used the toolkit developed by Zhang Le[4]. We set the maximum number of iterations to 500 in order to allow the parameters to convergence and the Gaussian penalty variance was set to 5.0 in order to avoid overfitting.

We performed the same text classification task as in Section 4.2 (RCV1 dataset, chronological split, *CCAT* class). In order to estimate the confidence of the classifier, we used the average of the entropies over the predictions on the test set. Since for binary classification tasks entropy is in the range of [0,1] and higher values indicate lower confidence, we used $1 - entropy(x)$ in order to be able to compare the curves directly with the ones of the margins produced by the SVM classifiers, in which higher margins indicate higher confidence. As in the previous experiments, 1% of the pool of unlabelled instances was added in each round of active learning and random selection was performed for comparison.

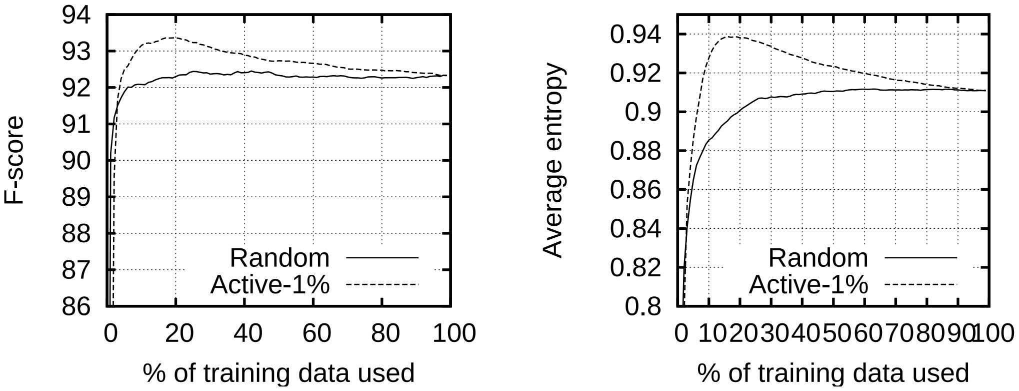

Fig. 6. Graphs for the performance (left) and the confidence (right) of Bayesian logistic regression for class *CCAT* of RCV1.

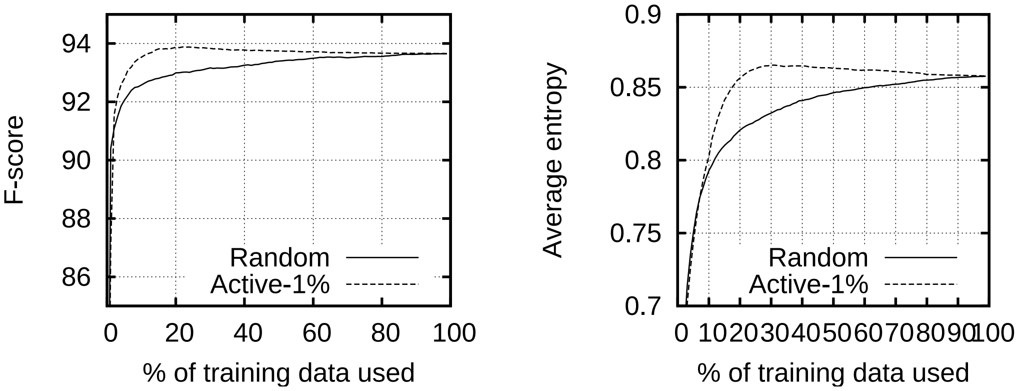

Fig. 7. Graphs for the performance (left) and the confidence (right) of maximum entropy for class *CCAT* of RCV1.

In the graphs of Figures 6 and 7 we can see the performance and the confidence curves for the BLR and the maximum entropy classifiers respectively

[4] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

during active learning and random selection. A first observation in both sets of experiments is that with a subset of the data selected with uncertainty-based sampling higher performance is achieved than with the whole dataset. In the case of BLR, the maximum performance achieved during active learning was 93.36% F-score at 20% of the data while the performance using the whole dataset was 92.75% F-score. In the case of maximum entropy the phenomenon was less pronounced, the highest F-score being 93.88% at 22% of the data and 93.65% F-score respectively. The same phenomenon was observed to a lesser extent in the SVM experiments of Section 4.2. The maximum performance achieved during AL with linear kernel SVMs was 94.08% F-score, while the performance achieved using all the data was 94.05%. Using Gaussian kernel SVMs, these performances were 94.65% and 94.64% respectively. Schohn and Cohn (2000) observed the same phenomenon in their active learning experiments with linear kernel SVMs and the Reuters-21578 dataset and attributed it to the noise from inconsistent annotation. While this is likely to be true to a certain extent, the consistency of the phenomenon suggests that it is related with the way data is selected during uncertainty-based sampling. As explained in Section 3, in the early stages uncertainty-based sampling avoids instances that contradict the knowledge gathered already by the classifier. Training on a smaller but unambiguous dataset (given the statistical learning algorithm and the feature representation), apart from resulting in higher confidence, could also achieve higher performance. However, further work is needed to verify this.

The confidence curves in the right graphs of Figures 6 and 7 confirm the applicability of the stopping criterion suggested in Section 3, since during uncertainty-based sampling a rise-peak-drop pattern is exhibited. In the case of BLR, a consistent drop starts when 20% of the data has been selected, which coincides with the peak of the performance. In the case of maximum entropy, a consistent drop starts when 30% of the data has been selected.

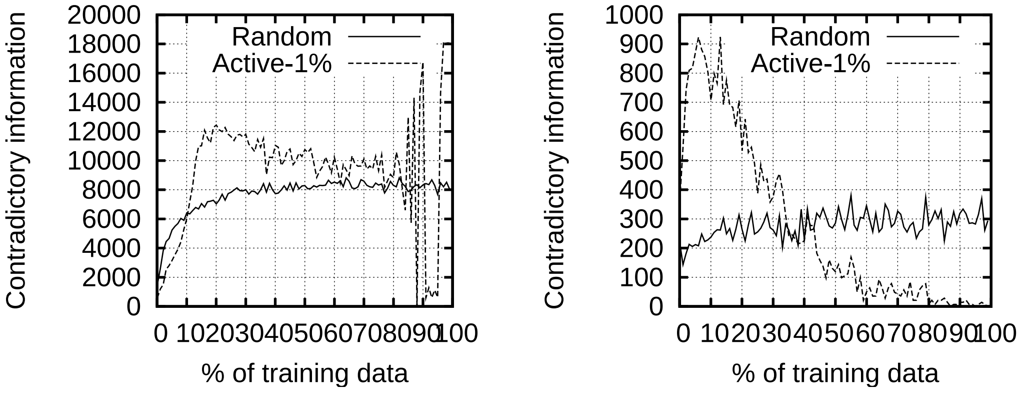

Fig. 8. Measuring the amount of contradictory information added in each round for Bayesian logistic regression (left) and maximum entropy (right).

In Figure 8 we plotted the amount of contradictory information added in each round during active learning and random selection for BLR and maximum entropy. As performed in Section 4.2 for the SVM classifiers, the amount of contradictory information contributed by each instance classified incorrectly was measured as the confidence over that instance divided by the average confidence of the classifier in that round. The calculation was the same in each round as in Equation 2, withe the only difference that since BLR and maximum entropy are probabilistic models and the task is binary, the confidence of the classifier was estimated as $1 - entropy(x)$ of the class distribution predicted instead of the decision margin which was used for SVMs. As with the SVM classifiers, it can be observed that initially the amount of contradictory information added during active learning in the early rounds is very little and it rises afterwards causing the drop in the confidence of the classifier. On the contrary, for random sampling it is evenly distributed. The fluctuation observed in the final rounds of active learning of BLR (Figure 8, left graph) is due to the fact that the model has become very confident and at that point it predicted the large majority of instances correctly or incorrectly in alternate rounds.

4.5 Named entity recognition experiments

In order to test the wider applicability of the stopping criterion suggested, we performed further active learning experiments with named entity recognition (NER). For this purpose, we used the data from the CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003), in which four entity types had to be recognized by the systems. Since the SVM active learning method described in Section 4.1 deals with binary classification task, we transformed the multiclass NER task into a binary one. We performed this by collapsing all the entity classes into a general entity class, thus reducing the task to classifying the tokens as being part of a named entity or not. The resulting dataset has a skewed class distribution, since 85% of the tokens do not belong to an entity. The sizes of the training and the test set are 203,621 and 51,362 tokens respectively.

The SVM classifier built uses simple lexical features, such as capitalization, the presence of digits and/or punctuation marks, as well as suffixes. Also, we used the part-of-speech tags (provided by the organizers) and the tokens themselves as features. In our experiments, we randomly chose 1% of the training data as seed data and the rest was used as the pool of unlabelled data. We used the test set provided by the task organizers to evaluate the performance measuring the F-score achieved. In each round, using the method described in Section 4.1, we chose a batch of tokens to be added to the training data. As with text classification, we used two different batch sizes, adding 1% and 0.1% of the pool data in each round and performed random sampling. Also, we tracked the confidence of the classifier by measuring the average margin on the test set. The results appear in Figure 9.

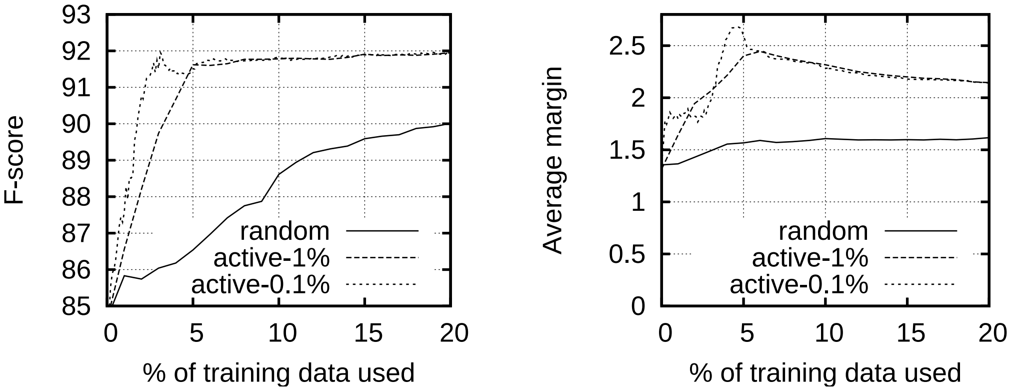

Fig. 9. Graphs for the performance (left) and the confidence (right) for linear kernel SVMs on NER.

The gains from applying SVM active learning to NER are impressive. When 1% of the pool selected in each round, the classifier achieves 90.7% F-score using 5% of the data, and when 0.1% of the pool selected in each round only 3% is needed for this level of performance. During random sampling, 32% of the data is required. Similar encouraging results were reported by Shen et al. (2004). The rise-peak-drop pattern of the confidence during active learning was observed as well (bottom graph of Figure 9), therefore the stopping criterion suggested in Section 3 is applicable. It is reasonable to expect that this should be the case in most natural language tasks, since the inherent ambiguity of natural language is very likely to generate instances that to a certain statistical model seem to be contradictory. Similar results were obtained using Gaussian kernel SVMs.

Comparing these results to those on text classification, an interesting point is that random selection seems to be less efficient for NER than it is for text classification and conversely, that active learning seems to be more beneficial in the case of NER. These differences can be attributed to the complexity of the annotation unit considered in each case. In text classification, a document represented using bag-of-words, even if randomly selected, is likely to contain some useful information for the classifier, since it is unlikely that all its words would have been encountered in the training data, especially in the early stages. In NER though, it is more likely to encounter the same or very similar tokens which do not contribute to the performance of the classifier. As a result, active learning, by selecting tokens on which the classifier is uncertain, is able to make more informative selections.

At this point it must be noted that there is always the risk that a misleading and/or noisy seed set or some misguided selections in the initial rounds can lead to incorrect separation boundaries, as well as non-optimal satisfaction of the stopping criterion. As observed in the case of the SVM active learning for NER adding 0.1% of the training data in each round, the confidence of the classifier drops temporarily without having exhausted the dataset. It is therefore advisable to allow for a consistent drop in the confidence to be observed, before considering the stopping criterion fulfilled.

5 Multiclass SVM active learning

In the experiments of Section 4.5 we applied active learning combined with support vector machines to a reduced version of the named entity recognition. The impressive results obtained motivated us to explore the possibility of tackling the multiclass task. We consider as multiclass tasks those in which each instance is assigned to exactly one class and the number of available classes is larger than two. An example of such a task is text classification on the 20 Newsgroups dataset (Lang, 1995), as opposed to the Reuters RCV-1 (Lewis et al., 2004) where each document can have multiple labels and so the task is commonly treated as a series of binary classification tasks. In the following sections, we briefly discuss multiclass SVM classification and we introduce a method for performing uncertainty-based sampling in this scenario.

5.1 Multiclass SVM classification

Support vector machines in their standard formulation are binary classifiers. Their success motivated researchers to investigate extensions that would allow multiclass classification with this model. Several methods have been presented for this purpose. The most popular of them decompose the multiclass task to several binary classification ones for training and combine the output of the binary classifiers during testing. A popular strategy for achieving this is the one-against-all scheme, in which for each class a separate classifier is trained against the rest of the data. During testing, the class whose classifier has the largest positive margin is selected (Vapnik, 1995). Another strategy is the one-against-one scheme in which binary classifiers are trained for each pair of classes and during testing voting among the classifiers takes place to decide on the class (Hsu and Lin, 2002). Other strategies involve error-correcting codes (ECOC) in order to reduce the multiclass task to binary ones (Rennie and Rifkin, 2001). It has was observed though that the performance of the combined multiclass classifier is more dependent on the performance of the binary ones, rather than the strategy used to combine them (Rennie and Rifkin, 2001).

5.2 Uncertainty-based sampling with multiclass SVMs

In order to perform uncertainty-based sampling, we used the one-against-all scheme described in Section 5.1. The choice was made due to its simplicity in implementation, as well as the fact that the scheme uses the margins of the classifiers directly for its decisions. For the purpose of uncertainty-based sampling, we need to define a measure of uncertainty for the decisions of the multiclass SVM classifier, which in turn will be used to select instances for labelling. In the one-against-all scheme, the class whose classifier has the largest positive margin is selected. To estimate the confidence of the classifier we used an idea from Schapire et al. (1997). They define the confidence of a multiclass classifier as the difference between the weight assigned to the correct label and the maximal weight assigned to any of the other labels. In order to adapt it to multiclass SVMs and be able to use it during active learning we had to make two alterations. First, since during active learning the correct labels are not available for the pool instances at the time they are selected for annotation, we consider the label decided by the one-against-all classifier instead. Second, we considered the margins of the classifier as the weights of the labels. More formally, given an instance x and an ensemble of binary SVM classifiers with decision functions $f_i(x, a_i)$ as defined in Equation 1, the confidence of the multiclass classifier $c(x)$ is:

$$\begin{aligned} i &= argmax_i f_i(x, a_i) \\ j &= argmax_{j \neq i} f_j(x, a_j) \\ c(x) &= f_i(x, a_i) - f_j(x, a_j) \end{aligned} \tag{5}$$

The way the confidence is estimated in Equation 5 assigns higher confidence to instances for which one binary classifier yields a large positive margin and all the others large negative margins. It is also worth noting that it reduces the confidence on instances for which there are more than one binary classifiers yielding positive margins, as well as when none of the classifiers yields a positive margin. The resulting estimate is always positive, ranging from 0 to ∞ and the higher the value the higher the confidence. As with the margins yielded from binary SVMs, these estimates are not comparable across different datasets and/or kernels. For uncertainty-based sampling, in each active learning round we select the instances with the lowest confidence $c(x)$.

5.3 Experiments

For the multiclass SVM active learning experiments two tasks were used, named entity recognition (Tjong Kim Sang and De Meulder, 2003) and shallow parsing (Tjong Kim Sang and Buchholz, 2000). In both cases, the implementation of SVM-Light with linear kernel and the default parameters were used in order to build multiclass classifiers using the one-against-all scheme. In each round, we measured the average confidence of the multiclass SVM classifier over the instances of the test set using Equation 5.

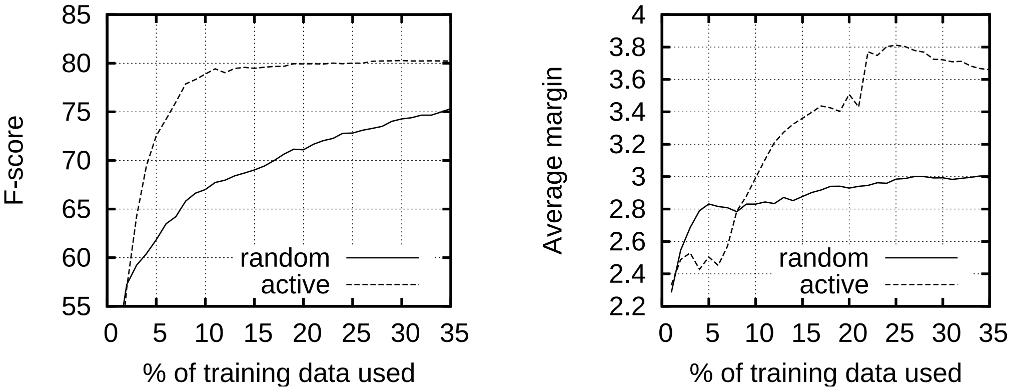

Fig. 10. Graphs for the performance (left) and the confidence (right) for the multiclass SVM classifier during active learning for the CoNLL-2003 NER task.

In Figure 10 we present the graphs for the performance (left graph) and confidence (right graph) of the multiclass SVM classifier during uncertainty-based sampling for the NER task. The experimental setup was the same as in the Section 4.5, except for the fact that the classes (person, location, organization and miscellaneous) were considered independently[5]. The savings achieved were substantial, reaching 80% F-score at 20% of the data used during uncertainty-based sampling using the method introduced in Section 5.2. The same performance level during random selection was achieved using 90% of the data. Also, the stopping criterion is applicable, even though there were fluctuations in the rise-peak-drop pattern. These can be attributed to the fact that the selections made in each round during AL take into account all the binary classifiers. Therefore, it is very likely that for a given binary classifier only some of the selections in each round will be informative for it. As a consequence, the pattern is not exhibited as distinctly as in the case of binary classification.

The purpose of shallow parsing, as defined by Tjong Kim Sang and Buchholz (2000), is to divide text into syntactically related non-overlapping groups of tokens (chunks). In our experiments, we used the data from the shared task of CoNLL 2000[6]. The sizes of the training and testing set are 211,727 and 49,389 tokens respectively. Each token belongs to one syntactic category, such

[5] The B-entity tags were used only for the first token of an entity immediately following a token of a different entity of the same class. As a result, they are very rare so we merged them with their respective I-entity tags in order to reduce the running time of our experiments.

[6] http://cnts.uia.ac.be/conll2000/chunking/

as verb phrase (VP) or noun phrase (NP). The corpus is annotated using the IOB tagging scheme and there are 11 syntactic categories, resulting in 23 classes. The number of instances in each class varies significantly, from 1 instance for the I-LST class to more than 63,000 instances for the I-NP class. In accordance with the shared task, we evaluated the performance measuring the F-score. Following Kudoh and Matsumoto (2000) who used SVMs for this task, the tokens themselves and their respective part-of-speech tags (which were provided by the organizers) from a 5-token window were used as features.

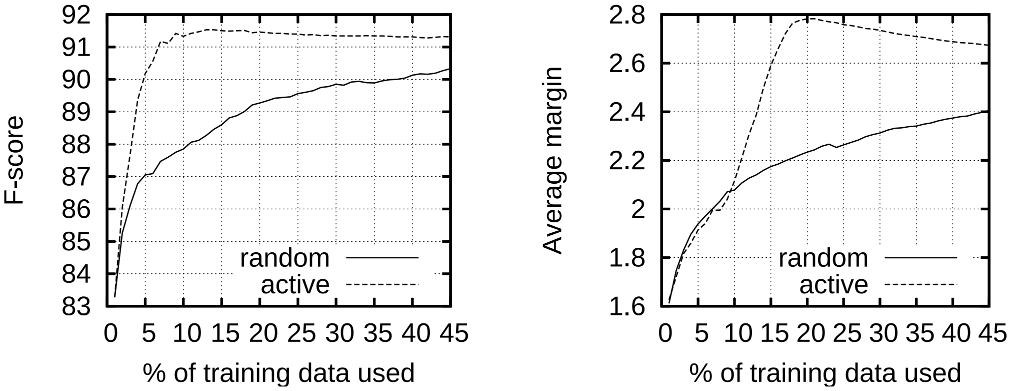

Fig. 11. Graphs for the performance (left) and the confidence (right) for the multiclass SVM classifier during AL for shallow parsing.

In the experiments shown in Figure 11, 1% of the training set randomly selected was used as initial training data and the rest was used as the pool of unlabelled data. Two runs were performed, one with active learning as described in Section 5.2 and one with random selection. In each round, 1% of the pool was added to the training data of the classifier. As the left graph demonstrates, during active learning the learning curve of the classifier is far steeper than during random selection. In particular, having used 10% of the data, the F-scores achieved were 91.33% and 87.85% respectively. The rise-peak-drop pattern of the confidence (right graph) is exhibited during active learning, but like in the case of the multiclass NER experiments there are fluctuations. Therefore, in practice a consistent drop in the confidence should be allowed in order to avoid premature termination.

6 Related Work - Discussion

The stopping criterion suggested in this paper requires a consistent drop in the confidence of the classifier for its fulfillment. Defining what constitutes a consistent drop, i.e. the number of rounds and/or the rate of drop compared to the maximum confidence reached is not straightforward. They are likely to depend on the task and the dataset used and in particular on the differences between instances that cannot be resolved by the combination of statistical model and feature representation used. For example, in the experiments with the imbalanced dataset (Section 4.3), the drop in the confidence was much slower than in the case of the balanced dataset of the experiments of Section 4.2. However, such differences are unlikely to be known beforehand when the dataset is unlabelled.

In practice, an operational stopping criterion based on our experiments would be to stop annotating when the confidence of the classifier drops for a few consecutive rounds. While there is no guarantee that this would always be the ideal stopping point since it could be a local maximum of the confidence curve, in our experiments the longest drop in the confidence of the classifier that occured before its global maximum was reached lasted two consecutive rounds [7]. Terminating the annotation when the confidence drops for three consecutive rounds yields the results of Table 1. Nevertheless, further investigation of this issue is an interesting direction for future work.

experiment	stopping point	perf. at stopping point	max perf
CCAT, linear, 1%	20%	93.98%	94.08%
CCAT, linear, 0.1%	15.2%	94%	94.08%
CCAT, Gaussian, 1%	25%	94.58%	94.62%
C16, linear, 1%	6%	49.98 %	51.03%
CCAT, BLR, 1%	24%	93.25%	93.36%
CCAT, MaxEnt, 1%	34%	93.79%	93.88%
NER, linear, 1%	9%	91.77%	92.31%
NER, linear, 0.1%	5%	91.51%	91.96%
NER-multi, linear, 1%	28%	80.24%	80.28%
chunking, linear, 1%	24%	91.40%	91.53%

Table 1
Stopping points and the performances achieved for the AL experiments of this paper.

An obvious way to estimate the performance and/or the confidence of the classifier would be to use the resulting labelled dataset after each round of AL. However, a dataset selected in this fashion is likely to contain only hard instances, especially if the active learning selection is successful (Baram et al., 2004). Using it for confidence estimation during active learning in particular would not be appropriate because in each round new instances are added with different difficulty.

[7] In the experiments of Section 4.5 when selecting 0.1% of the instances in each round and in the NER experiments of Section 5.3

Several active learning stopping criteria that do not require a labelled dataset have have been proposed in the literature. Schohn and Cohn (2000) suggest that data labelling should cease when there are no instances in the pool that lie closer to the separating hyperplane than the support vectors that define it. In our experiments, with linear kernel SVMs applied to text classification for the Reuters RCV1-v2 *CCAT* class and adding 1% of the pool to the training data in each round, using this criterion we would stop after having added 24% of the available data. Applying the stopping criterion suggested in this paper we would have terminated active learning when 20% of the data has been used. The performance would have been roughly the same, approximately 94% in F-score. In the case of binary NER adding 1% of the pool in each round, the criterion of Schohn and Cohn (2000) is fulfilled at 11% of the data and the performance at that point was 91.83% in F-score. While both criteria are satisfied at roughly the same point, the stopping criterion of Schohn and Cohn (2000) is based on an observation specific to binary SVM classifiers which restricts its applicability. Our criterion is based on an observation on how uncertainty-based sampling selects the data and the nature of the data itself, therefore it can be generalized beyond SVMs. Campbell et al. (2000) suggested the same criterion adding a subsequent evaluation step on a randomly selected and manually labelled dataset, in which a human uses the evaluation to judge whether the performance of the classifier is satisfactory.

More recently, Zhu and Hovy (2007) suggested stopping when the entropy of each selected unlabelled instance is below a certain threshold and the classifier can predict the labels of these instances correctly. However, it is not straightforward how to specify the value for the entropy threshold. Also, the entropies vary according to the number of classes and in tasks in which the latter is not constant (for example in parse selection), they cannot be used directly. Moreover, if we consider non-probabilistic classifiers, the margins are not comparable for different classifiers and/or datasets and therefore it is unlikely that a certain threshold value would be applicable in all cases. We attempted to verify this stopping criterion in our experiments with Bayesian logistic regression which is a probabilistic method (Section 4.4) but it proved to be ineffective because there were no rounds in which the labels of all the selected instances were predicted correctly. In the case of the SVM classifier with the linear kernel (Section 4.2), this condition of the criterion was fulfilled only after 96% of the data has been annotated, which is much later than the 20% point at which we would stop using the stopping criterion suggested in this work. A possible reason for this is that the batch size used in our experiments was significantly larger (1607 instances compared to 10 in the experiments of Zhu and Hovy (2007)) and it is less likely that the classifier can predict the labels of all the instances in the batch correctly.

In other related work, Tomanek et al. (2007) conducted active learning experiments using the query by committee approach and they found that the disagreement rate between the classifiers of the committee can be used as a stopping criterion. In particular, they suggest that annotation should cease after the point at which the classifiers of the committee stop disagreeing. We could not apply this criterion in our experiments, since we concentrated on using a single classifier to select data using uncertainty based sampling. Shen et al. (2004) performed active learning with SVMs for multiclass named entity recognition. In their work, they considered criteria other than uncertainty in the way they selected instances. However, they trained independent classifiers for each entity class, therefore not attempting mutually exclusive multiclass classification.

Another related issue is the reusability of the data. Baldridge and Osborne (2004) found in their experiments that reusing material selected during active learning with a different classifier and/or feature set is not very effective and can yield worse results than random selection. However, recent findings by Tomanek et al. (2007) show that reusability is feasible to a certain extent. While our work does not deal with this issue directly, the definition of a stopping criterion minimizes the potential cost of annotating data that is not reusable.

Concerning the named entity recognition and shallow parsing experiments, it must be noted that selecting tokens independently of each other is unlikely to be a realistic simulation of how a human annotator would deal with the task, since in many cases the context is needed to determine the class of a token. We used this experiment in order to test the applicability of the stopping criterion. Hachey et al. (2005) suggest that sentences are a more realistic annotation unit for NER. The choice of annotation unit and the way the annotation cost is estimated would affect our experiments, as shown in Ngai and Yarowsky (2000) and Baldridge and Osborne (2007), and it is an interesting direction for future work. Another issue related with the NER and shallow parsing tasks is that in recent years the use of sequential models such as Conditional Random Fields (Sutton and McCallum, 2006) has become the standard approach employed. Testing the applicability of of the stopping criterion suggested in this work with such models exceeds the scope of this work. It is worth pointing out that the selections made by the current selection model using SVMs could not be used to train sequential models because the latter require sentences completely annotated as training material. This observation supports the concerns about the reusability of the data expressed by Baldridge and Osborne (2004).

Finally, another issue related to our work is how active learning selections affect the performance of the human annotators, which is directly related to estimating the actual cost of annotation. Baldridge and Osborne (2004) measured the number of decisions made by the humans to select the correct parse in their experiments with parse selection. Hachey et al. (2005) studied the interaction of active learning with the inter-annotator agreement on the selected instances and observed that the instances selected with active learning are harder to annotate, resulting in annotation inconsistencies and noisy training data. In Ngai and Yarowsky (2000) it was shown that the performance of the trainable system correlates strongly with that of the annotator who created the training data used.

7 Conclusions - Future Work

The main contribution of this paper is the definition of a stopping criterion for active learning using uncertainty based sampling which does not require annotated data. We verified its applicability in two NLP tasks (text classification and named entity recognition) using suppport vector machines (with linear and Gaussian kernel), maximum entropy models and Bayesian logistic regression. Such a stopping criterion can be very useful when applying statistical NLP techniques to new domains where there is a paucity of annotated material. In addition, we presented a method for performing uncertainty-based SVM active learning for multiclass tasks using the one-against-all formulation and its efficiency was demonstrated in two tasks, named entity recognition and shallow parsing. Also, the applicability of the stopping criterion suggested was verified in these experiments.

Future work should study the relation of the stopping criterion with the reusability of the data, which is a very important issue. In particular, it would be of interest to study the conditions under which data annotated during active learning can be reused efficiently. Moreover, it would be interesting to see if a similar stopping criterion could be applicable in the case of domain adaptation. Finally, experiments employing human annotators are of interest in order to assess the results of active learning more realistically.

Acknowledgements

This work started under the supervision of Miles Osborne at Edinburgh University during the MSc thesis of the author and was continued under the supervision of Ted Briscoe at the University of Cambridge. The author would like to thank Mark Craven, Zoubin Ghahramani, Bobby Grammacy and the anonymous reviewers for helpful discussion and comments on this work.

References

Agichtein, E., Gravano, L., June 2000. Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM International Conference on Digital Libraries. Association for Computing Machinery, San Antonio, TX, USA, pp. 85–94.

Argamon-Engelson, S., Dagan, I., 1999. Committee-based sample selection for probabilistic classifiers. Journal of Artificial Intelligence Research 11, 335–360.

Baldridge, J., Osborne, M., July 2004. Active learning and the total cost of annotation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain, pp. 9–16.

Baldridge, J., Osborne, M., 2007. Active learning and logarithmic opinion pools for HPSG parse selection. Natural Language Engineering 13 (1), 1–32.

Baram, Y., El-Yaniv, R., Luz, K., 2004. Online choice of active learning algorithms. Journal of Machine Learning Research 5, 255–291.

Becker, M., Osborne, M., 2005. A two-stage method for active learning of statistical grammars. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence. Edinburgh, Scotland, pp. 991–996.

Berger, A. L., Pietra, S. D., Pietra, V. J. D., 1996. A maximum entropy approach to natural language processing. Computational Linguistics 22 (1), 39–71.

Buckley, C., Salton, G., Allan, J., 1994. The effect of adding relevance information in a relevance feedback environment. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer-Verlag New York, Inc., New York, NY, USA, pp. 292–300.

Burges, C. J. C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2 (2), 121–167.

Campbell, C., Cristianini, N., Smola, A., 2000. Query learning with large margin classifiers. In: Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp. 111–118.

Chang, C.-C., Lin, C.-J., 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Cohn, D. A., Atlas, L., Ladner, R. E., 1994. Improving generalization with active learning. Machine Learning 15 (2), 201–221.

Genkin, A., Lewis, D. D., Madigan, D., 2006. Large-scale bayesian logistic regression for text classification. Technometrics.

Hachey, B., Alex, B., Becker, M., June 2005. Investigating the effects of selective sampling on the annotation task. In: Proceedings of the Ninth Conference on Computational Natural Language Learning. Ann Arbor, Michigan, pp. 144–151.

Hsu, C.-W., Lin, C.-J., 2002. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13 (2), 415–425.

Joachims, T., 1998a. Making large-scale support vector machine learning practical. In: B. Schölkopf, C. Burges, A. S. (Ed.), Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, MA.

Joachims, T., April 1998b. Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning. Vol. 1398 of Lecture Notes in Computer Science. Springer, pp. 137–142.

Keerthi, S. S., Lin, C.-J., 2003. Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15 (7), 1667–1689.

Kim, J. D., Ohta, T., Tateisi, Y., Tsujii, J., 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. In: Bioinformatics. Vol. 19, Suppl. 1. Oxford University Press, pp. 180–182.

Kudoh, T., Matsumoto, Y., September 2000. Use of support vector learning for chunk identification. In: Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E. (Eds.), Proceedings of the Fourth Conference on Computational Natural Language Learning. Lisbon, Portugal, pp. 142–144.

Lang, K., July 1995. Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning. Morgan Kaufmann, pp. 331–339.

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., Vapnik, V., 1995. Comparison of learning algorithms for handwritten digit recognition. In: Fogelman, F., Gallinari, P. (Eds.), Proceedings of the International Conference on Artificial Neural Networks. pp. 53–60.

Lewis, D. D., Yang, Y., Rose, T. G., Li, F., 2004. RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397.

Li, M., Sethi, I. K., 2006. Confidence-based active learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (8), 1251–1261.

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

Morgan, A. A., Hirschman, L., Colosimo, M., Yeh, A. S., Colombe, J. B., 2004. Gene name identification and normalization using a model organism database. Journal of Biomedical Informatics 37 (6), 396–410.

Ngai, G., Yarowsky, D., October 2000. Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong, pp. 117–125.

Pereira, F., Tishby, N., Lee, L., June 1993. Distributional clustering of English words. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics. Columbus, Ohio, USA, pp. 183–190.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Smola, A., Bartlett, P., Schoelkopf, B., Schuurmans, D. (Eds.), Advances in Large Margin Classifiers. pp. 61–74.

Rennie, J. D. M., Rifkin, R., 2001. Improving multiclass text classification with the Support Vector Machine. Tech. Rep. AIM-2001-026, Massachusetts Insititute of Technology, Artificial Intelligence Laboratory.

Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., July 1997. Boosting the margin: a new explanation for the effectiveness of voting methods. In: Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufmann, pp. 322–330.

Schein, A., Ungar, L., 2004. Optimality for active learning of logistic regression classifiers. Tech. Rep. MS-CIS-04-07, University of Pennsylvania Department of Computer and Information Science.

Schohn, G., Cohn, D., 2000. Less is more: Active learning with support vector machines. In: Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp. 839–846.

Seung, H. S., Opper, M., Sompolinsky, H., July 1992. Query by committee. In: Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory. ACM Press, pp. 287–294.

Shen, D., Zhang, J., Su, J., Zhou, G., Tan, C. L., July 2004. Multi-criteria-based active learning for named entity recongition. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, pp. 21–26.

Sutton, C., McCallum, A., 2006. An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (Eds.), Introduction to Statistical Relational Learning. MIT Press.

Tjong Kim Sang, E. F., Buchholz, S., September 2000. Introduction to the CoNLL-2000 shared task: Chunking. In: Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E. (Eds.), Proceedings of the Fourth Conference on Computational Natural Language Learning. Lisbon, Portugal, pp. 127–132.

Tjong Kim Sang, E. F., De Meulder, F., 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Daelemans, W., Osborne, M. (Eds.), Proceedings of the Conference on Computational Natural Language Learning. Edmonton, Canada, pp. 142–147.

Tomanek, K., Wermter, J., Hahn, U., June 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In: Proceedings of the Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Language Learning. Prague, Czech Republic, pp. 486–495.

Tong, S., Koller, D., 2001. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research 2, 45–66.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, N.Y.

Zhu, J., Hovy, E., June 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In: Proceedings of the Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Language Learning.

Prague, Czech Republic, pp. 783–790.