Semi-Automatic Grading of Students' Answers Written in Free Text

Nuno Escudeiro[1,2], Paula Escudeiro[1,3] and Augusto Cruz[1]
[1]Instituto Superior de Engenharia, Instituto Politécnico do Porto, Portugal
[2]LIAAD INESC Porto L.A., Portugal
[3]GILT, Porto, Portugal
nfe@isep.ipp.pt
pmo@isep.ipp.pt
1040266@isep.ipp.pt

Abstract: The correct grading of free text answers to exam questions during an assessment process is time consuming and subject to fluctuations in the application of evaluation criteria, particularly when the number of answers is high (in the hundreds). In consequence of these fluctuations, inherent to human nature, and largely determined by emotional factors difficult to mitigate, it is natural that small discrepancies arise in the ratings assigned to similar responses. This means that two answers with similar quality may get a different grade which may generate inequities in the assessment process. Reducing the time required by the assessment process on one hand, and grouping the answers in homogenous groups, on the other hand, are the main motivations for developing the work presented here. We believe that it is possible to reduce unintentional inequities during an assessment process of free text answers by applying text mining techniques, in particular, automatic text classification, enabling to group answers in homogeneous sets comprising answers with uniform quality. Thus, instead of grading answers in random order, the teacher may assess similar answers in sequence, one after the other. The teacher may also choose, for example, to grade free text answers in decreasing order of quality, the best first, or in ascending order of quality, starting to grade the group of the worst answers. The active learning techniques we are applying throughout the grading process generate intermediary models to automatically organize the answers still not fixed in homogeneous groups. These techniques contribute to reduce the time required for the assessment process, to reduce the occurrence of grading errors and improve detection of plagiarism.

Keywords: text mining, active learning, free-text assisted grading

1. Introduction

The assessment of free text answers is a demanding process requiring for a great effort from the evaluators, particularly when the number of answers to assess is high – in the hundreds. This process demands for high concentration levels and for long periods of time leading to fluctuations in the evaluator's level of concentration and mood.

Moreover, evaluating answers in random order or, at least, without any order related to the quality of the answer itself can lead to situations in which different grades can be assigned to two answers of similar quality. Evaluating an answer of average quality after evaluating a set of answers of much lower quality may lead the evaluator to inflate the grade assigned to the average answer. Similarly, when that same average answer is evaluated after evaluating a number of very high quality answers, may trigger the opposite effect which will result in grading the average answer lower than if it was assessed in other circumstances. Two answers of similar quality, assessed after a series of high quality and low quality answers, will get different grades with some probability. These effects may be more evident when the evaluation process is longer.

These effects could be reduced if the selection of the next answers to access is made judiciously – to have the answers of similar quality being evaluated sequentially – and by ascending or descending order of quality – to avoid sharp fluctuations in the emotional context and the mood of the evaluator.

We believe it is possible to standardize the application of assessment criteria for evaluating free text answers by applying text mining techniques. In particular, using automatic text classification, to identify the different levels of answers' quality and to group answers according to those levels, may reduce mood fluctuation in the evaluators. These techniques allow (a) reduce the time needed to complete the evaluation process (b) reduce the occurrence of errors of evaluation and (c) better detect cases of plagiarism.

Reference this paper as: Escudeiro, N, Escudeiro, P and Cruz, A. "Semi-Automatic Grading of Students' Answers Written in Free Text" *The Electronic Journal of e-Learning Volume 9 Issue 1 2011, (pp15-22), available online at www.ejel.org*

Our current work applies active learning techniques and automatic text classification to organize free text answers to exam questions in homogeneous groups and then to sort them according to their quality.

This process requires the evaluator (user, in general) to provide a pre-labeled set of answers based on which an automatic classification model will be built. The labels or classes represent the grades to assign to every answer. To generate these classification models we rely on Support Vector Machines (SVM) with a linear kernel. SVM (Chapelle et.al., 2006) classifiers are recognized as one of the most suited for text classification tasks.

This classification model is then applied to all answers that have not yet been assessed assigning an automatic label (rate) to them. The grouping of responses in homogeneous groups and their ranking will be based on these provisional ratings assigned automatically.

To generate a classification model that can recognize all relevant rates we need to obtain pre-labeled answers from each of these groups. These are previously labeled by the user. This initial phase – until you can obtain a classification model that covers all rates of relevance and that is sufficiently accurate – is more demanding and more prone to assessment errors. It is therefore important to make this a short-term phase, requiring the evaluator to evaluate only a small number of answers until all grades have been identified and characterized through these pre-labeled examples. To minimize the effort required for this initial phase and, simultaneously, shorten this learning phase we use a strategy of active learning called D-Confidence (Escudeiro et.al., 2009).

Active learning techniques select the most informative answers, given the known evidence, avoiding asking the evaluator to rank responses from low value added. This is how active learning reduces the workload required from the evaluator to build an appropriate classification model. Experimental results show that it is possible to obtain valid classification models and improve the assessment processes.

The remaining sections of this paper present a brief review of the area of active learning, in Section 2 and describe the D-Confidence algorithm in Section 3. Section 4 describes our proposal for assisted rating of free text answers. In Section 5 we present the evaluation that has been performed and Section 6 states our conclusions and future work.

2. Active learning

There are several paradigms suited to automatic classification with numerous applications.

The paradigm of *supervised learning* allows you to specify arbitrary concepts, specific to a given problem. However, it requires a set of fully labeled data which can be prohibitive when tagging cases is a costly process, as is usually the case with text documents. In addition, it demands for exemplary instances, previously labeled, for all classes to learn.

The *semi-supervised* classification (Chapelle et.al., 2006) allows the specification of particular needs without requiring a prior process of intensive labeling. Nevertheless, it also requires a minimum set of pre-labeled cases covering all the classes to learn.

Unsupervised algorithms require no prior labeling, however, this paradigm does not give any chance for the user to guide the model towards specific needs and there is no a priori guarantee that the groups automatically generated without any input from the user are aligned with the groups of interest to the user and the current problem at hand.

Active learning techniques which seem more suited to our purposes select the next case to label wisely, prompting the user for its label. The most informative case will be selected by the learning algorithm rather than being randomly selected as in the case in supervised learning. It is expected that a lower number of labels is required to achieve the same accuracy when compared to the fully supervised learning setting.

Active learning approaches (Angluin, 1988; Cohn et al., 1994; Muslea et al., 2006) reduce label complexity – the number of queries that are necessary and sufficient to learn a concept – by analyzing unlabeled cases and selecting the most useful ones once labeled. Queries may be artificially generated (Baum, 1991) – the *query construction* paradigm – or selected from a pool (Cohn

et al., 1990) or a stream of data – the query filtering paradigm. Our current work is developed under the query filtering approach.

The general idea in active learning is to estimate the value of labeling one unlabeled case. Query-By-Committee (Seugn et al., 1992), for example, uses a set of classifiers – the committee – to identify the case with the highest disagreement. Schohn et al. (2000) worked on active learning for Support Vector Machines (SVM) selecting queries – cases to be labeled – by their proximity to the dividing hyperplane. Their results are, in some cases, better than if all available data is used to train. Cohn et al. (1996) describe an optimal solution for pool-based active learning that selects the case that once labeled and added to the training set, produces the minimum expected error. This approach, however, requires high computational effort. Previous active learning approaches (providing non-optimal solutions) aim at reducing uncertainty by selecting the next query as the unlabeled example on which the classifier is less confident (Lewis and Gale, 1994).

Batch mode active learning – selecting a batch of queries instead of a single one before retraining – is useful when computational time for training is critical. Brinker (2003) proposes a selection strategy, tailored for SVM, that combines closeness to the dividing hyperplane – assuring a reduction in the version space close to one half – with diversity among selected cases – assuring that newly added examples provide additional reduction of version space. Hoi et al. (2006) suggest a new batch mode active learning relying on the Fisher information matrix to ensure small redundancy among selected cases. Li et al. (2006) compute diversity within selected cases from their conditional error.

Dasgupta (2005) defines theoretical bounds showing that active learning has exponentially smaller label complexity than supervised learning under some particular and restrictive constraints. This work is extended in Kaariainen (2006) by relaxing some of these constraints. An important conclusion of this work is that the gains of active learning are much more evident in the initial phase of the learning process, after which these gains degrade and the speed of learning drops to that of passive learning.

Agnostic Active learning (Balcan et al., 2006), A^2, achieves an exponential improvement over the usual sample complexity of supervised learning in the presence of arbitrary forms of noise.

This model is studied by Hanneke (2007) setting general bounds on label complexity.

All these approaches assume that we have an initial labeled set covering all the classes of interest.

Clustering has also been explored to provide an initial structure to data or to suggest valuable queries. Adami et al. (2005) merge clustering and oracle labeling to bootstrap a predefined hierarchy of classes. Although the original clusters provide some structure to the input, this approach still demands for a high validation effort, especially when these clusters are not aligned with class labels.

Dasgupta et al. (2008) propose a cluster-based method that consistently improves label complexity over supervised learning. Their method detects and exploits clusters that are loosely aligned with class labels.

Among other paradigms, it is common that active learning methods select the queries which are closest to the decision boundary of the current classifier. These methods focus on improving the decision functions for the classes that are already known, i.e., those having labeled cases present in the training set. The work presented in this paper diverts classifier attention to other regions increasing the chances of finding new labels.

In this work, we used the active learning algorithm called D-Confidence. This approach affects the learning process and labeling diverting it to regions of space where not yet explored. Thus minimizing the number of questions you need to do to find cases representing all classes.

The D-Confidence selects cases based on a tagging feature that adds confidence that the current classifier has a particular class - a traditional criterion in active learning - the distance between this case and the previously known cases of that class by the classifier.

This criterion is biased for cases that do not belong to known classes - lower confidence - and that are located in regions of space where unexplored - Distance high to known classes. This way is more efficient in the even coverage of the concepts to learn.

Common techniques of active learning based on a low confidence of the classifier instantiated to select cases to label (Angluin 1988) and assume that the chaos pre-labeled cover all the classes to learn - this assumption is not valid in our case. These approaches use the classification model from each iteration to calculate the confidence in each class for each unlabeled case, then selecting the unlabeled case with the least confidence.

Other more recent approaches make use of unsupervised classification to find cases to label (Dasgupta et.al, 2008).

3. D-Confidence

The most common active learning approaches rely on classifier confidence to select queries (Angluin, 1988), assume that the pre-labeled set covers all the labels to learn and are focused on accuracy. Our scenario is somehow different: we do not assume that we have labeled cases for all classes and, besides accuracy, we are mainly concerned with the fast identification of representative cases from all classes. To achieve our goals we rely on the d-Confidence selection criterion (Escudeiro et al., 2009), which is effective in the early detection of exemplary cases from unseen classes. Instead of relying exclusively on classifier confidence we propose to select queries based on the ratio between classifier confidence and the distance to known classes. D-Confidence, weighs the confidence of the classifier with the inverse of the distance between the case at hand and previously known classes.

D-Confidence is expected to favor a faster coverage of case space, exhibiting a tendency to explore unseen regions in case space. As a consequence, it provides faster convergence than confidence alone. This drift towards unexplored regions and unknown classes is achieved by selecting the case with the lowest d-Confidence as the next query. Lowest d-Confidence is achieved by combining low confidence – probably indicating cases from unknown classes – with high distance to known classes – pointing to unseen regions in the case space. This effect produces significant differences in the behavior of the learning process. Common active learners focus on the uncertainty region asking queries that are expected to narrow it down. The issue is that the uncertainty region is determined by the labels we known at a given iteration. Focusing our search for queries exclusively on this region, while we are still looking for exemplary cases on some labels that are not yet known, is not effective.

Unknown classes hardly come by unless, by chance, they are represented in the current uncertainty region.

In active learning, the learner is allowed to ask an oracle (typically a human) to label examples – these requests are called *queries*. The most informative queries, given the goals of the classification task, are selected by the learning algorithm instead of being randomly selected as is the case in passive supervised learning. The general idea in active learning is to estimate the value of labeling one unlabeled case. The general algorithm below is the basement for active learning classification.

Initialize U_0, L_0
Train the learner in L_0 *to generate* h_0
Run h_0 *to classify instances in* U_0
$i = 1$
While not stopping criteria {
    Select $q_i = \langle x_i \rangle$ *pertencentea* U_{i-1}, *the most adequate queries from* U_{i-1}
    Ask the oracle for their labels, c_i
    $L_i = L_i \cup \langle x_i, c_i \rangle$
    $U_i = U_i$ *minus* $\langle x_i, c_i \rangle$
    Train the learner in L_i *to generate* h_i
    Run h_i *to classify instances in* U_i
    i++
}

D-Confidence is a particular algorithm to select queries which is particularly suited to reduce labeling effort at an initial stage of the classification process. This initial stage, when the user is still trying to find representative cases from all the grades he is interested in, is the most critical in our current work.

D-Confidence is based on the ratio between confidence and distance among cases and known classes (Eq. 1).

Eq.1 $$dConf(u) = \max_k\left(\frac{conf(c_k \mid u)}{median(dist(u, xlab_k))}\right)$$

For a given unlabeled case, *u*, the classifier generates the posterior confidence w.r.t. known classes. Confidence is then divided by an indicator of the distance between the unlabeled case at hand and all labeled cases belonging to class *k*, $xlab_k$. This distance indicator is the median of the distances between the unlabeled case at hand and all labeled cases belonging to the class. We expect the median to soften the effect of outliers. D-Confidence for each known class is computed by dividing class confidence for a given case by the aggregated distance of the unlabeled case to that class. Finally, we compute d-Confidence of case *u*, *dConf(u)*, as the maximum d-Confidence on individual classes.

4. Assisted grading of free text answers

To evaluate the gains of our approach, we have developed a prototype (Figure 1) for assisted assessment and grading of free text answers to examinations questions.

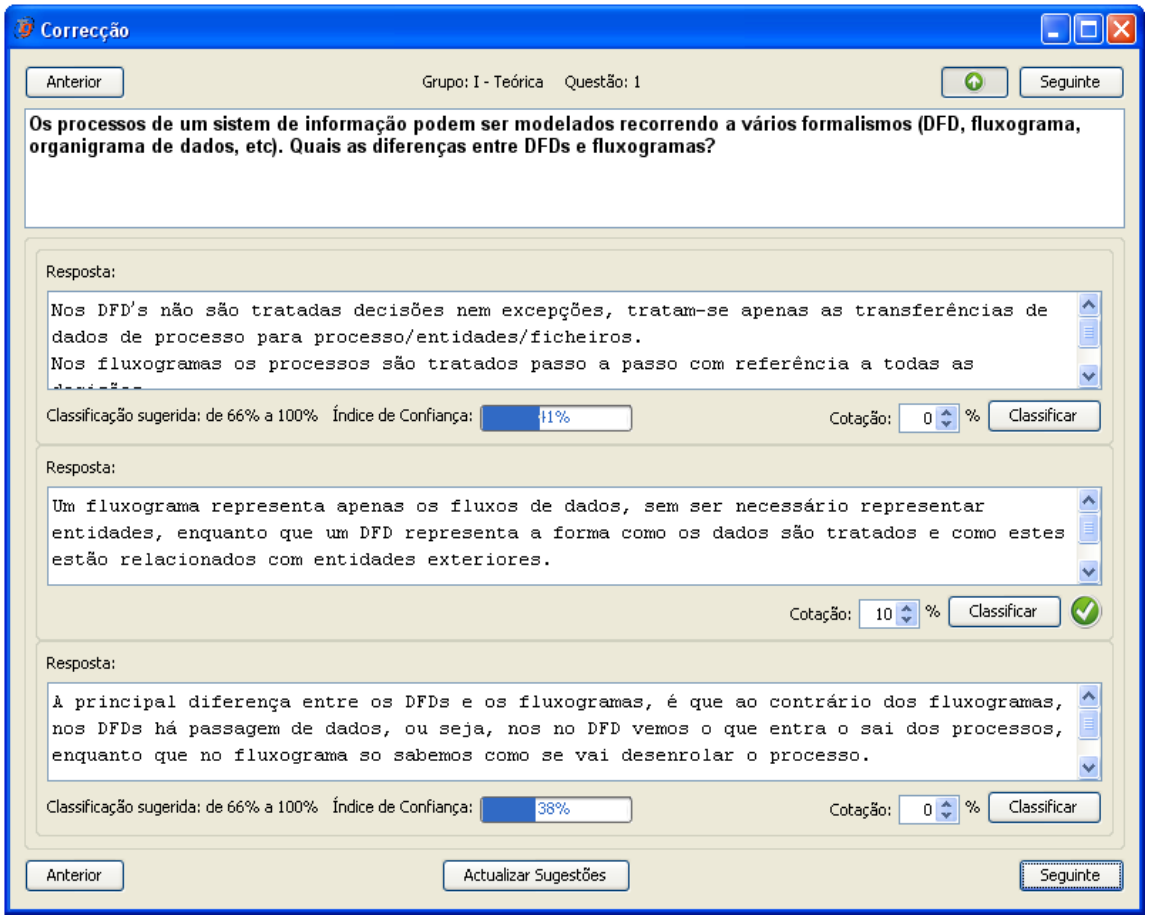

Figure 1: Learning process, evaluation and grading of answers to train the automatic classification model

This prototype compiles the free-text answers to a given question and pre-processes them into a model that is suited for the application of automatic classification techniques (pre-processing). In particular, the set of answers to a given question is converted into a TFxIDF matrix (Weiss 2005 et.

al) that represents, each answer, a text document, by a row in this matrix. The columns, in turn, represent the terms or, more generally, the elements of natural language that occur in the answers. The techniques of automatic text classification are then applied to this array of vectors representing the corpus of the students' answers to a given question.

The assessment process is initiated by the user (teacher) that analyzes and classifies a set of answers (a minimum of two is required). These will be the seed labeled cases required to kick off d-Confidence. In our prototype, the ratings given by the teacher should be referred to the scale 0-100. These answers, previously labeled by the teacher, are used to train a SVM classifier (the learning task). This classifier generates a classification model for a set of labels, or classes, representing grades. The maximum number of different grades is automatically set by the prototype based on the number of answers in the set being assessed. This limitation stems from the fact that we need a minimum number of examples for each class so that we can ensure some accuracy in automatic classification.

Depending on the number of answers, *N*, to a certain question, we set the maximum number of classes (grades), *K*, that the classifier is able to learn using Eq.2:

Eq. 2 $$K = \max\left(3, \frac{N}{10}\right)$$

The number of classes to learn will always be between *1* and *K*. When assessing students' answers, if the teacher assigns a grade of *C* (in the scale 0-100) then, for the purpose of organizing the homogeneous groups, this answer will be assigned to class k_i – the corresponding grade to C that our classifier can distinguish with a certain confidence. Equation 3, is used to define the class to which an answer graded by the teacher with a mark of *C* belongs to:

Eq.3 $$k_i = \text{int}\left(\frac{C \times K}{100}\right) + 1$$

Thus, in a case with 25 answers to evaluate, for instance, we will have them grouped into three groups or classes/grades. The first class groups students' answers rated by the teacher with a score between 0% and 33%, class two, groups the answers graded between 34% and 66% and class three, groups the answers rated between 67% and 100%.

Once the number of different classes mapping teacher grades is set and the first classifier is generated, from a set of pre-labeled answers, the automatic learning process runs in parallel with the grading process until the moment when the teacher finds that the automatic classification model is reasonable.

Every iteration of the learning process begins by applying the current classification model to all the answers that have not yet assessed. This model assigns a class/grade to each un-assessed answer with a given confidence.

After these classes are assigned to each answer the D-Confidence active learning algorithm is applied to sort the unlabeled answers and to present them to the teacher in descending order of their informative value to the current classification model, i.e., by increasing order of their d-Confidence measure.

The teacher evaluates the answer with greater informative value, which is then added to the set of all the answers previously evaluated. This set of teacher-labeled answers is then used to generate a new classification model – which is expected to be more accurate because it was trained on the basis of more cases properly labeled – and the process iterates.

These iterations are repeated until the classification model can estimate grades for un-assessed answers in a manner deemed sufficiently accurate by the teacher. Thereafter this lastly generated classification model is applied to group the answers that have not been assessed yet in groups of

homogeneous quality joining together the answers that have been assigned to the same grade by the classification model.

These homogeneous groups are then presented to the teacher by ascending or descending order of their quality according to the preference of the teacher.

5. Evaluation

The evaluation of our proposal was based on a real dataset, with 31 free text students' answers to a question from an examination of the course of Software Engineering.

As the number of responses is 31 we have considered three classes corresponding to the grades in the ranges 0-33%, 34-66% and 67-100%, according to Eq.2 and Eq.3.

All answers have been assessed by the teacher that has assigned a grade between 0 and 100 to every answer. These teacher grades are hidden from the classifier. They will be used for evaluation purposes only to compare automatic grades to those assigned by the teacher and so compute the accuracy of the automatic classifier.

The accuracy of the classifier is calculated on the set of answers that were not used to train the current classifier.

In the first iteration only three quotations from different classes where provided to the classifier. The classification model generated from these three labeled cases was applied to the remaining 28 student's answers (those that have not been used for training) assigning a grade to each one. From these classes automatically assigned by the classifier, 6 correspond to the true class of the answer, as assessed by the teacher, which corresponds to an accuracy of 21% (6/28). In this first experiment, the classifier was trained based on three cases and does not have an acceptable accuracy. Confidence is low and the teacher may not accept the suggestions given by the classifier.

In a second iteration 6 labeled answers were provided to the classifier, two for each different grade. The percentage of correct predictions in all the answers that were not used to train the classifier is 36% (9 hits in 25). In this iteration the confidence index is still low, and the teacher may not accept the suggestions of the classifier.

In the third iteration, nine labels were submitted, three labeled answers from each grade. The percentage of correct predictions is 45% (10/22).

Finally, in a fourth iteration we have provided 12 graded answers to the classifier, four from each grade. The percentage of correct predictions is now 68% (13/19).

Although an accuracy of 68% is not a high standard, it seems reasonable for the intended purpose. We must recall that the purpose of automatic classification in our current work is not to make the automatic attribution of grades to the students' answers, but simply to allow grouping answers in homogeneous groups to reduce errors in the assessment process and to reduce the time required by the assessment process.

6. Conclusions

Grouping responses of similar quality, those that are worth similar grades, prior to having them assessed by the teacher, may reduce errors arising from a random assessment order constantly interspersing answers with high quality with others of lesser quality.

For the assessment process to be done this way it is necessary to have some semi-automatic way of creating these homogeneous groups and of sorting them by ascending or descending order of their quality. This stems from the effort that, otherwise, would necessarily be required from the teacher to organize students' answers in these groups, probably would make the whole process useless.

Our evaluation, although not too extended, provides interesting results and indicates that, despite a poor accuracy, it is possible to create homogeneous groups of answers using techniques of automatic text classification.

One reason for the low accuracy probably relates to the fact that the answers are all on the same topic. This, being essential for such an application, reduces the usefulness of the classifier. A test made with a corpus on different topics has achieved 96% of correct suggestions (the suggestion referred to the theme that the document concerned). Under these circumstances (the need to distinguishing text documents all referring to the same topic) we are studying the potential impact on the accuracy of the classifier of the use of other models of representation of the answers besides TFxIDF. We are examining the benefits of using models based on multi-term attributes (N-grams), models focused in certain categories of words (part-of-speech tagging) and string kernels (Lodhi 2002).

References

Adami, G, Avesani,P and Sona, D. Clustering documents into a web directory for bootstrapping a supervised classification. Data and Knowledge Engineering, 54:301–325, 2005.

Angluin, D, Queries and Concept Learning, Machine Learning, 2, 319-342, 1998

Balcan, M.-F., Beygelzimer, A, and Langford, J. Agnostic active learning. In In ICML, pages 65–72. ICML, 2006.

Baum, E. Neural net algorithms that learn in polynomial time from examples and queries. IEEE Transactions in Neural Networks, 2:5–19, 1991.

Brinker, K. Incorporating diversity in active learning with support vector machines. In Proceedings of the Twentieth International Conference on Machine Learning, 2003.

Chapelle, O, Schoelkopf, B and Zien, A, (ed.) Semi-supervised Learning, MIT Press, Cambridge, MA, 2006

Cohn, D, Atlas, L, and Ladner, R. Training connectionist networks with queries and selective sampling. In Advances in Neural Information Processing Systems, 1990.

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

Cohn, D, Ghahramani, Z, and Jordan, M. Active learning with statistical models. Journal of Artificial Intelligence Research, 4:129–145, 1996.

Dasgupta, S. Coarse sample complexity bonds for active learning. In Advances in Neural Information Processing Systems 18. 2005.

Dasgupta, S. and Hsu, D. Hierarchical sampling for active learning, In Proceedings of the 25th International Conference on Machine Learning, 2008

Escudeiro, N and Alípio, J, Efficient coverage of case space with active learning. In P. M. L. M. R. Lus Seabra Lopes, Nuno Lau, editor, Progress in Artificial Intelligence, Proceedings of the 14th Portuguese Conference on Artificial Intelligence (EPIA 2009), volume 5816, pages 411–422. Springer, 2009

Hanneke, S. A bound on the label complexity of agnostic active learning. In Proceedings of the 24th International Conference on Machine Learning, 2007.

Hoi, S, Jin, R, and Lyu, M. Large-scale text categorization by batch mode active learning. In Proceedings of the World Wide Web Conference, 2006.

Kaariainen, M. Algorithmic Learning Theory, chapter Active learning in the non-realizable case, pages 63–77. Springer Berlin / Heidelberg, 2006.

Lewis, D, Gale, D and W. A. A sequential algorithm for training text classifiers. In SIGIR ’94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc., 1994.

Liu, H and Motoda, H. Instance Selection and Construction for Data Mining. Kluver Academic Publishers, 2001.

Muslea, I, Minton, S, and Knoblock, C, A. Active learning with multiple views. Journal of Artificial Intelligence Research, 27:203–233, 2006.

Lodhi, H, Saunders, C, Shawe-Taylor, J, Cristianini, N and Watkins, C, Text Classification Using String Kernels, Journal of Machine Learning Research, 2, 419-444, 2002

Schohn, G and Cohn, D. Less is more: Active learningwith support vector machines. In Proceedings of the International Conference on Machine Learning, 2000.

Seung, H, Opper, M, and Sompolinsky, H. Query by committee. In Proceedings of the 5th Annual Workshop on Computational Learning Theory, 1992.

Weiss, S, Indurkhya, S, Zhang, T, Damerau, F, Text Mining, Predictive Methods for Analyzing Unstructured Information, Springer, 2005

