# Active Learning for the Text Classification of Rock Climbing Logbook Data

Eoghan Cunningham[1,2] and Derek Greene[1,2]

[1] School of Computer Science, University College Dublin, Ireland
[2] Insight Centre for Data Analytics, University College Dublin, Ireland

**Abstract.** This work applies active learning to the novel problem of automatically classifying user-generated logbook comments, published in online rock climbing forums. These short comments record details about a climber's experience on a given route. We show that such comments can be successfully classified using a minimal amount of training data. Furthermore, we provide valuable insight into real-world applications of active learning where the cost of annotation is high and the data is imbalanced. We outline the benefits of a model-free approach for active learning, and discuss the difficulties that are faced when evaluating the use of additional training data.



## 1 Introduction

The purest style of traditional rock climbing is called "on-sight" climbing. This refers to climbing a route with no more information than can be seen from the ground[3]. Any information about a rock climb that would aid a climber in their ascent is known as "beta'. Beta often takes the form of tips or instructions about how to climb the route and what equipment to bring. It is typical for climbers to share beta with each other before attempting a route. However, many climbers who seek a greater challenge and hope to claim an "on-sight" ascent, will actively avoid reading or hearing beta prior to their ascent.

Rock climbers in the UK and Ireland currently log ascents of routes on forums hosted by UKClimbing.com (UKC). When logging ascents, UKC users can leave comments about their experience on a given route. The site has recently added functionality which allows users to label the comments that contain beta information. Like spoilers in a film, some users wish to avoid reading beta before climbing a route. With the addition of such labels, these users are able to hide comments that contain beta. However, the vast majority of the comments on UKC remain unlabelled. In this work we investigate the potential for a text classification algorithm to be used to automatically assign labels to these comments.

On collecting a dataset of comments from UKC, we found that the scenario here corresponds to that of many supervised machine learning tasks, where an abundance of unlabelled data exists, but the amount of training data available is quite limited. A description of this novel dataset is provided in Section 3.1.

[3] https://en.wikipedia.org/wiki/Glossary_of_climbing_terms

Crowdsourced annotations have proven to be useful when creating new training sets from unlabelled natural language corpora [13], but this process requires considerable time and effort on the part of the human annotators.

Therefore, in this work we employ *active learning* methods [15] to compile a larger training set, allowing us to build a text classification model that generalises well to unseen comments. In active learning, the examples chosen for human labelling are selected carefully, so as to yield the maximum increase in classification accuracy with the minimum amount of human effort. This project investigated the effectiveness of a range of different text classification methods and active learning strategies, as applied to the large dataset of comments collected from UKC. These experiments are described in Section 3.2.

After comparing alternative active learning approaches, we adopt the model-free, exploration based strategy EGAL [5] to build an informative training set. As our training set would be labelled by a committee of expert human annotators, the cost of annotation was particularly high. This represents a unique case where the benefits of a model-free approach are especially important, as we see in Section 4.1. Using the additional training data, we are able to improve the performance of our comment classifier. The evaluation of the quality of this additional training data is a key challenge of this work, as outlined in Section 4.3. Finally, we achieve further improvements by including non-textual comment metadata in the classification process, as discussed later in Section 4.4.

## 2 Related Work

### 2.1 Active Learning

In many supervised learning tasks, there are not enough labelled samples to create sufficient training and test sets. Often the cost of labelling samples to create such sets is very high. In these cases *active learning* can be employed to reduce the number of samples that need to be labelled.

In typical supervised learning tasks, training data is a random or stratified sample selected from the larger space. In the case of active learning, the learner builds the training set by choosing the samples which are most useful to the classification process. This is to reduce the number of samples that need to labelled. The key hypothesis is that, if a learning algorithm is allowed to choose the data from which it learns, it will perform better with less training [12]. *Pool-based active learning* was shown to reduce the amount of training data required to achieve a given level of accuracy in a text classification task 500-fold [7]. Given a pool of unlabeled data $U$, an active learner is made up of the following: a classifier $f$ trained on a pool of labeled data $X$, and a query function $q()$. Given the data in the labeled pool $X$, $q()$ decides which instances in $U$ to query next. The samples queried from $U$ are presented to an *oracle*, typically a human annotator, to be labelled and added to $X$. The query function $q()$, often called the *selection strategy*, is a vital component of an active learning system.

In the work by Lewis and Gale [7], the query function selected the unlabeled samples about which the classifier was least certain. This approach is known as

*uncertainty sampling*. A query function performing uncertainty sampling in a binary classification task may simply return the unlabeled samples for which the probability of positive classification is closest to 0.5 [7, 6]. Settles and Craven proposed an *information density* framework [12], which is based on the idea that the most informative queries are not only those which are uncertain, but also those that are the most representative of the distribution of the samples in the data. Such samples are found in dense regions of the sample space. The information density approach will weight the uncertainty of a sample by the its average similarity to other samples in the sample space.

As methods that consider only density and/or uncertainty can fail to explore the sample space, some selection strategies will seek to select queries that are differ from the samples in the labelled pool. Exploration guided active learning (EGAL) [5] is a selection strategy which uses both diversity and density to find informative queries. By querying from dense regions of the space, EGAL avoids querying outliers while ensuring that selected samples are suitably diverse from the labelled pool of samples. Further, as EGAL does not measure uncertainty in sampling, it can be implemented without a classification model. This model-free approach allows for all of the queries to be found and annotated in a single batch. In many applications where human annotation is not readily available, model-free approaches are highly favorable over model-based approaches in which models are retrained after each smaller batch [10].

### 2.2 Crowdsourcing Annotated Data

There has been much success in recent years using crowdsourcing to annotate data. *reCaptcha* is one of the best known examples of crowdsourcing in classification tasks [16]. To determine if users were computers or human, *captchas* were introduced as an "automated public Turing test", where the user was asked to decipher some text. reCaptchas introduced a second similar task. In this case the text used is not known in advance, but rather is typically scanned text from a corpus of text that is being digitized. If a human user provides a satisfactory answer to the captcha, their response is considered to be a useful annotation for the original scanned image. More recently, sites like Amazon's *Mechanical Turk* [11] have been a popular source for human-annotated data. Mechanical Turk (MTurk) has been shown to be effective in collecting annotations for a range of classification tasks [14]. However, when domain-specific expertise is required, MTurk is often not a suitable source of information.

## 3 Methods

In this section we describe the key methods used to identify the most appropriate active learning approach for our data, how we applied this approach to collect additional training data and finally, how we evaluated the resulting augmented training set. We also include a brief description of the dataset we have collected.

### 3.1 Dataset

A dataset of 100,000 comments was collected from UKC.com in September 2019. These include ascents covering the period 1952–2019, and the average comment was just 15 words long. In addition to the comment text, we also collected additional non-textual features associated with each comment. It was hypothesised that these additional features might aid in comment classification. In the case of each comment, where the information was available, the location of the route, the location that the climber was local to and the maximum grade achieved by the climber that year were collected as comment metadata. Of the 100,000 comments collected, 304 (0.304%) of them were labelled as containing *beta*. On inspecting those comments and their labels, an expert human annotator agreed with only 170 of them. It is clear that the data source contains even fewer labelled comments than initially expected and that the quality of this labelling is very poor. However, manual re-annotation of these comments provided a labelled pool of comments that we were confident were correctly labelled and could serve as a starting point for the active learning process. While 99.7% of the comments in this dataset remain without class labels, we believe there to be a large class imbalance in the data where comments containing *beta* are in the minority. We include two comments from the dataset below to provide insight into the nature of the data and the language used to give beta. Comment 1 contains beta, while Comment 2 does not.

1. "Laybacked the main crack until the prominent right foothold, then stepped up with right foot whilst reaching up with left hand for a sound hold."
2. "Damp when we first arrived but soon brightened up. An excellent afternoon, first visit to the crag."

A cross-validation experiment conducted on the pool of 304 annotated comments showed that the classification of *beta* was a learnable task, but that a larger training would be required to improve classification performance and ensure the model would generalise well to unseen data. Our dataset may prove useful to other researchers, in particular those considering classification of user generated text, short text or datasets with a large class imbalance.

### 3.2 Active Learning Selection Strategies

To expand our training set as efficiently as possible, we sought to find an appropriate active learning strategy for our dataset. We evaluated six different experimental combinations: three selection strategies, each with two classification models. The models proposed were a Support Vector Machine (SVM) with a linear kernel and a Random Forest classifier as these were shown to perform well for our data. The selection strategies were uncertainty sampling, random sampling, and EGAL. In each case the comments were represented as TF-IDF vectors of uni-grams, bi-grams and tri-grams.

In order to evaluate the different approaches, we simulated active learning on our pool of labelled samples. A small random labelled pool was removed and used as a training set while the remaining data was considered unlabelled. On

each iteration, samples would be selected from the unlabelled pool, added to the training set and used to train a classification model. As we lacked sufficient annotated data to remove a hold-out test set, at each iteration, the model was evaluated on all of the available annotated data. Each selection strategy was then evaluated by plotting the classification performance (balanced accuracy score) over the size of the training set. The best performing approach would be the one that produced the steepest learning curve, ie. the approach that achieved the highest accuracy with smallest amount of training data. As each model was tested on a test set that contained samples also present in the training set, the resulting accuracy would likely overestimate the performance on unseen data, but the overall shape of the learning curve would still be informative. This approach to evaluation is common when the labelled dataset is small [8]. While using a hold out test set, or even testing on the unlabelled pool, are also common approaches [1, 2], they are only feasible in cases were a fully-annotated dataset is available.

In each case, the learning process was repeated 10 times, each with a different labelled pool of 10 randomly-chosen samples. All base labelled pools were balanced with 5 positive and 5 negative samples. In each case classification was evaluated using a balanced accuracy score which was averaged over the 10 experiments. The results of this experiment are provided in 4.1.

### 3.3 Annotation Environment

The selection strategy that performed best in the experiment outlined above was used to query 100 comments from the un-labelled of comments. These comments were labelled by a committee of expert annotators and added to the labelled comments used for training. A web page to developed to allow for remote, reliable annotation of comments by the human annotators. A brief pilot study was conducted with three participants in order to evaluate this annotation tool. Participants were asked to log on, annotate 30 comments and complete a questionnaire about their experience. From this pilot study we asserted that the site was functioning properly and that we could expect annotators to label 100 comments over the course of a week.

After collecting the user judgements from the annotation environment, the labels were evaluated to assess inter-annotator agreement and quantify consensus. In addition to a simple percentage agreement calculation, a Cohen's kappa statistic [3] was calculated. Following this, majority voting was used to assign a class label to every comment. Each annotator could also be ranked by the rate at which they disagreed with this majority. This was done to identify unreliable annotators. These annotators could then be removed from the study, and consensus and agreement measures can be recalculated. This is based on the greedy consensus rules proposed by Endriss and Fernandez [4], where annotators were only permitted to disagree with the majority vote $t$ times before being removed from consideration. The results of these calculations are outlined in Section 4.2.

### 3.4 Augmented Training Set

The active learning stage of the project resulted in 100 additional samples that had been queried by our learner and labelled by participants using the annotation environment. In this section we outline the experiments used to evaluate the quality of this additional training data, and investigate how it might improve our ability to perform text classification of comments from UKC.com. The additional training samples were evaluated using a hold out test set of the comments most recently posted to UKC. Two models were trained; a pre-active learning model, trained on the original pool of labelled comments and a post-active learning model, trained on the original comments and the additional training data. Using our unseen test set, we could evaluate how each classifier might perform in a real-world deployment scenario. In our evaluations we report both a Balanced Accuracy Score (BAS) and the Area under the ROC Curve (AUC). The BAS is used due to the class imbalance in the data and reports performance on an unseen test set, at a decision threshold chosen after parameter tuning. The AUC instead considers performance across all decision thresholds and the classifier that maximises AUC is considered to be better able to distinguish between the two classes. While cross-validation is a common approach to evaluating classification, particularly in cases where annotated training and test data is scarce, we deemed it inappropriate for evaluating our additional training data for the following reasons:

1. The additional training data should not be used as test data, as this would reward any selection strategy that queried samples similar to the already labelled training data and may result in trivial classification.
2. The original training data should not be used as test data as the additional training samples are chosen to augment the training set, not to replace it.

Instead, we have used cross-validation only as a means of parameter-tuning. In the case of each model, a 10 fold cross-validation was performed using all of the data available to the model, i.e., the pre-active learning model was tuned using only the samples that had been labelled prior to active learning.

The test set was composed of 150 comments that had been most recently posted to UKC.com. These were annotated by three expert human annotators. We refer to this set as Set 1. This set was heavily imbalanced, as it contained only 17 samples that were labelled as *beta*. While this test set was representative of the typical comments posted on UKC, a more informative test set should contain more instances of the positive class. Although it may have been possible to find more examples of *beta* by searching for key words or other textual features, it was deemed that this was likely to result in trivial classification.

We chose instead to use features not included in the classification process to attempt to find more instances of the *beta* class. From our initial labelled set of comments, we found that there were some users who were very active on the site and were more likely to include beta in their comments. By collecting up to 15 of the most recent comments from 10 such users we collected a set of 139 comments which, on annotation, was shown to contain 37 positive samples. This final set was added to the original test set to give a larger, richer test

set of 289 comments. We refer to this set as set 2. Both pre- and post-active learning models were tuned, trained and evaluated on each set and the results are reported in Section 4.3.

### 3.5 Additional Features

In addition to the textual features, three non-textual features were extracted and evaluated. (i) *local_to*, a boolean features indicating if the climber was local to the area where the route was climbed. (ii) *comment_length*, the length of the comment. (iii) *challenge*, a feature describing how challenging the climber would find the route, calculating by comparing the grade of the route to the max grade achieved by the climber in the same year. In the case of each non-textual feature, the impact of that feature on the target variable was measured using information gain. This was calculated using the annotated base set and the additional data from the EGAL queries. The unseen test set was omitted from these calculations. The effect of including these features in classification was then evaluated using the post-active learning training set and the final unseen test set. Given the relatively small number of additional features, it was possible to evaluate all possible combinations of features exhaustively. For each subset of features, the model was trained on all of the textual features and that subset. Performance on the test set was reported as BAS and AUC.

## 4 Results

### 4.1 Selection Strategy

The results of the experiment comparing different active learning selection strategies are provided below. Table 1 reports the area under the learning curve for each selection strategy, up to a training set size of 100 samples. Figure 1 shows the learning curves of the best performing selection strategies: (i) uncertainty sampling using a random forest classifier; (ii) EGAL using a random forest classifier. The smoother learning curve achieved using uncertainty sampling is due to the larger batch size. Each model-based approach was implemented using a batch size of 10, while EGAL was implemented with a batch size of 1.

As outlined in Section 2.1, EGAL is a model-free approach, meaning the classifier does not need to be retrained between batches of queries. As a result, it can be implemented with a batch size of 1 and still, all 100 queries can be made before any annotation is received. This is not the case however for model-based approaches like uncertainty sampling. In the case of uncertainty sampling, each batch must be annotated before the next batch can be queried. In light of this, the batches used to evaluate uncertainty sampling may have been too small. It is anticipated that, even with a larger batch size of 20 samples, the extent of annotation that was received would have been reduced, since annotators could not provide labels as and when they wished. The next batch of queries could not be made until the previous batch had been fully annotated.

| Selection Strategy | AUC (0-100) |
|---|---|
| Random Sampling (SVM) | 60.41 |
| Uncertainty Sampling (SVM) | 57.75 |
| EGAL (SVM) | 49.92 |
| Random Sampling (RF) | 55.88 |
| Uncertainty Sampling (RF) | **68.41** |
| EGAL (RF) | 67.09 |

**Table 1.** AUC scores scores for different query selection strategies.

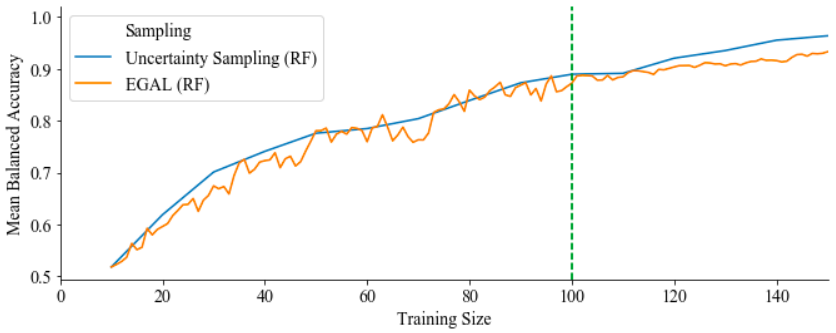


**Fig. 1.** Learning curves for the best performing active learning strategies.

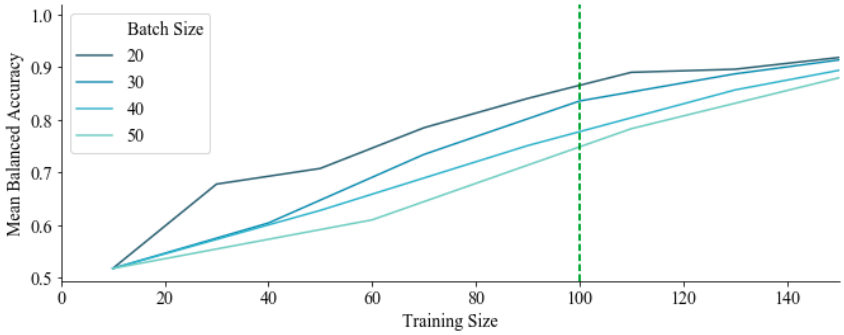


**Fig. 2.** Learning curves for uncertainty sampling, with different batch sizes.

Figure 2 plots the learning curves achieved using uncertainty sampling with a random forest classifier, as the batch size is increased. It is clear from this figure that for larger batches, the classification performance decreases. EGAL is shown to outperform the other strategies across all batch sizes larger than 10. In general it is desirable to use larger batches to reduce dependencies amongst annotators and increase the total number of annotations that we receive. For this reason we have selected EGAL for our active learning system.

### 4.2 User Annotation

Using EGAL, 100 comments were queried from the pool of unlabelled comments and annotated by 10 human experts using our annotation tool. Here we report the results of some brief experiments to evaluate consensus amongst these annotators. In total we received 680 labels, meaning on average comments would be assigned a class label aggregated over close to 7 votes. Using pairwise agreement we calculated a total agreement percentage of 76.7% and a Cohen's kappa value of 0.498. Percentage agreement is often criticized as a means of inter-annotator agreement as it does not account for chance agreement. If we assign 10 random labels to each of our 100 comments, we achieve a 50% percentage agreement. For this reason we consider Cohen's kappa in value in addition to percentage agreement. A kappa value between 0.4 and 0.6 is considered moderate agreement.

We also chose to consider the rate at which annotators disagreed with the majority. The average annotator disagreed with the majority between 8 and 12% of the time. However there we two outliers; annotators who disagreed with the majority 21 and 33% of the time. These annotators received kappa scores of 0.21 and 0.32 indicating very low agreement. After removing these annotators from the study we recalculated the agreement measures. Across the remaining annotators the percentage agreement was calculated to be 82% with a kappa of 0.63. This indicated substantial agreement.

On investigation it was found that both of the unreliable annotators had strong biases that lead to their poor agreement scores. 17 of the 18 errors made by one annotator were false positives, while the other problematic annotator made 17 false negatives. It is encouraging to find that the unreliable annotators were biased and not simply inconsistent. Were these annotations very inconsistent, it may suggest that labelling beta is not a learnable task. The fact that they are seen to be biased suggests that, with better calibration, their agreement scores could have been improved. This calibration could be achieved by changing the instructions and examples provided to the annotators prior to annotation.

### 4.3 Augmented Training Set

Table 2 reports the balanced accuracy score (BAS) and area under the ROC curve (AUC) for each model on each of the unseen test sets. We can see that in all cases the classifier trained post active learning outperforms the classifier trained before active learning. The balanced accuracy scores are achieved using the decision threshold selected from the cross validation and as such, represent our best estimate of the models performance were it to be deployed. The AUC reported is the area under the ROC curve and instead quantifies the classifiers performance across all decision thresholds. This may provide a more robust means of comparing the models as it is not subject to decisions made in the cross-validation process, namely: choosing the best decision threshold. This choice of decision threshold may be overfitting to our relatively small labelled set used in the cross-validation. While, there may be a better choice, that better generalizes to all comments, the balanced accuracy scores reported in table 2 represent our best effort at labelling unseen data both before and after active learning.

| Metric | Pre-Active Learning | Post-Active Learning |
|---|---|---|
| AUC Set 1 | 0.794 | **0.824** |
| BAS Set 1 | 0.699 | **0.715** |
| AUC Set 2 | 0.830 | **0.837** |
| BAS Set 2 | 0.736 | **0.756** |

**Table 2.** AUC and BAS scores on unseen test sets.

| Feature | Information Gain |
|---|---|
| challenge | 0.0175 |
| is_local | 0.0015 |
| comment_len | 0.1825 |
| Textual Features | |
| 'right' | 0.0790 |
| 'reach' | 0.0686 |
| 'crack' | 0.0672 |
| 'foot' | 0.0604 |
| 'left' | 0.0584 |

**Table 3.** Feature Information Gain.

| Feature Set | AUC | BAS |
|---|---|---|
| comment_len | 0.850 | 0.795 |
| is_local | 0.843 | 0.765 |
| challenge | 0.844 | 0.772 |
| comment_len, is_local | 0.851 | 0.780 |
| comment_len, challenge | 0.849 | 0.781 |
| is_local, challenge | 0.839 | 0.765 |
| comment_len, is_local, challenge | 0.845 | 0.760 |

**Table 4.** AUC and BAC scores for additional features on unseen test set.

### 4.4 Additional Features

Each of the additional non-textual features were evaluated by calculating their information gain on the target class. These values are reported in table 3. The five most informative textual features are included in this table for comparison. When we consider these textual features we see examples of the highly specific language that is common to the comments containing beta. Words like 'reach', 'left', 'right', and 'foot' are all indicative of the instructive language used to give beta. We can see how these terms form the instructions that might help a climber succeed on a route. Table 3 shows the comment length to be the most informative, not only of the additional features but of any feature.

In addition to calculating the information gain, it was also proposed to evaluate the additional features by including them in the classification and assessing their effect on performance. Given the relatively small number of additional features, it was possible to evaluate each possible combination of features. Table 4 reports the performance of the post-active learning classifier on the final unseen test set with each combination of the additional features. These results show that the addition of any of the non-textual features resulted in an increase in performance. However, unsurprisingly the comment length feature proved to be the most beneficial, achieving a 4% increase in BAS and a 1.3% increase in AUC.

In an attempt to better understand the effect of these non-textual features on comment classification, it was decided to repeat these evaluations using our pre-active learning classifier. Figure 3 shows how the non-textual features effect classification performance in both pre- and post-active learning models. Each

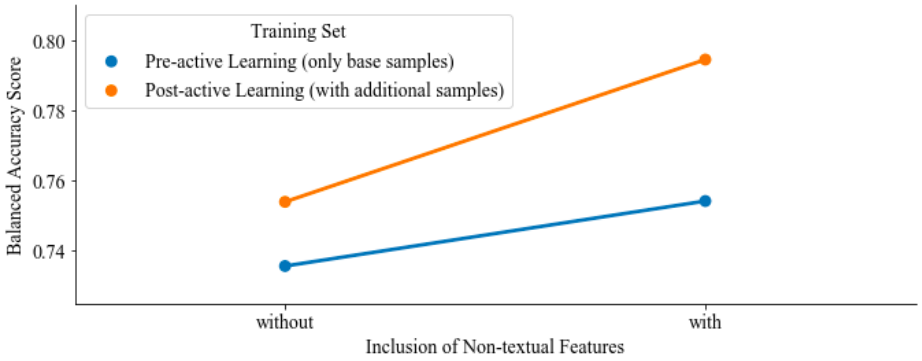


**Fig. 3.** Illustration of the effect of additional features on classification.

line in figure 3 reports the BAS before and after the non-textual features were included. The slope of the lines represent the effect of these on the model's performance. It is apparent from the figure that the pre-active learning model sees less improvement from the additional features than the post-active learning model. This is a further testament to the quality of the active learning training data. In essence, with this additional training data, our model makes better use of the non-textual features.

## 5 Conclusion

We sought to apply active learning to the problem of automatically classifying user-generated logbook comments from online rock climbing forums. UKC.com implemented labels on their site which allow users to identify comments containing *beta* yet, the majority of these comments remain unlabelled. We have shown our final classification model can identify *beta* in comments on UKC.com with 80% accuracy. We believe this to be close to the upper limit of what is achievable for this classification task. As we saw n Section 4.2, expert human annotators achieve 82% pairwise agreement in the same task. We have used Exploration Guided Active Learning (EGAL) to identify the most informative samples to add to our training data. By including non-textual metadata features in the classification process, we have improved our model's balanced accuracy on an unseen test set by $\approx 6\%$. We believe the techniques we have employed should prove useful in similar tasks involving the identification of anomalies in user-generated text where annotation is limited. An obvious analog to our work is identifying spoilers in movie reviews. Further, we have highlighted the necessity for the use of model-free active learning approaches in cases where the cost of annotation is high. Many model-based selection strategies are evaluated using batches of five or fewer samples [7, 9, 15], as this offers the best results in off-line experiments. However, such an approach is not feasible when annotations are crowd-sourced. Finally, we provided insights into the complications associated with evaluating active learning techniques when the cost of annotation is high.

The new dataset compiled for this work is made available online[4], and may prove useful to other researchers, particularly those considering classification of short, user-generated text or data with a large class imbalance.

**Acknowledgement.** This work was supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2.

## References

1. Baldridge, J., Osborne, M.: Active learning and the total cost of annotation. In: Proc. Conf. Empirical Methods in Natural Language Processing. pp. 9–16 (2004)
2. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: Proc. 20th Int. Conference on Machine Learning. pp. 59–66 (2003)
3. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1), 37–46 (1960)
4. Endriss, U., Fernández, R.: Collective annotation of linguistic resources: Basic principles and a formal model. In: Proc. 51st Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 539–549 (2013)
5. Hu, R., Delany, S.J., Mac Namee, B.: EGAL: Exploration Guided Active Learning for TCBR. In: Proc. 18th International Conf. Case-Based Reasoning Research and Development. pp. 156–170. Springer-Verlag (2010)
6. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proc. Int. Conf. Machine Learning. pp. 148–156. Morgan Kaufmann (1994)
7. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proc. 17th Annual International ACM SIGIR Conf. Research and Development in Information Retrieval. pp. 3–12. Springer-Verlag New York (1994)
8. Mac Namee, B., Delany, S.J.: Sweetening the data set: Using active learning to label unlabelled datasets. In: Proc. 19th Irish Conference on Artificial Intelligence and Cognitive Science (2008)
9. McCallumzy, A.K., Nigamy, K.: Employing em and pool-based active learning for text classification. In: Proc. Int. Conf. on Machine Learning). pp. 359–367 (1998)
10. O'Neill, J., Delany, S., MacNamee, B.: Model-Free and Model-Based Active Learning for Regression, vol. 513, pp. 375–386 (01 2017)
11. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using Amazon's Mechanical Turk. In: Proc. NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk
12. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
13. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.: Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In: Proc. Conf. Empirical Methods in Natural Language Processing. pp. 254–263 (2008)
14. Sorokin, A., Forsyth, D.A.: Utility data annotation with amazon mechanical turk. IEEE Conf. Computer Vision and Pattern Recognition Workshops pp. 1–8 (2008)
15. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Machine Learning Research **2**, 45–66 (2002)
16. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: recaptcha: Human-based character recognition via web security measures. Science **321**(5895), 1465–1468 (2008)

[4] https://github.com/eoghancunn/logbookdataset