Improving Text Classifier Performance through Human-in-the-Loop Error Correction: Enhancing Learning from Explanations

Xinyang Song
Supervised by: Edwin Simpson

June 21, 2023

Abstract

Text classification models work with natural language processing to analyse text and assign labels. In crisis scenarios such as floods and earthquakes, text classifiers can be used to identify and forward emergency text reports from social media to relevant agencies. However, the effectiveness of text classifiers relies heavily on a large amount of labelled training data, which can be scarce and difficult to obtain [1]. In addition, training large amounts of labelled data can delay model response times, and unrepresentative data can affect model accuracy. Identifying actionable types of information, such as search and rescue requests, remains challenging.

The project integrates real and generated simulated user judgements and explanations into the classifier training process, using an active learning strategy for multi-classification problems to optimise the human-in-the-loop (HITL) process for error correction as an alternative method to address these limitations. In conjunction with a previous technique, Representation Engineering with Natural Language Explanations (ExpBERT), this technique adds feature generation from explanations combined with original features to improve the classifier's performance [2]. The main goal of this project is to exceed the maximum achievable accuracy of the ExpBERT text classifier by employing multiple modalities of interactive systems in the classification process [3]. Highly informative unlabelled instances are queried during the iterative process using sampling strategies based on uncertainty, representativeness and diversity corresponding to the aspects. Active learning is trained with a small amount of valid data, which reduces the labelling effort and processing time in the text classification task and allows for an accurate explanation of the text.

Explore the effectiveness of active learning on the system. The annotator accepts and processes the extracted completed unlabelled instances for training. The valid data generated through this process with labels and accurate explanations as well as optimal hyperparameters will be used to fine-tune the ExpBERT classifier's explanation generation process as well

as the training process, repeatedly iterating to improve its accuracy eventually. To evaluate the performance of this method, this experiment will set up a comparison group comparing ExpBERT classification models using different active learning strategies, models with and without Bayesian active learning improvements and models under other algorithms on a CrisisNLP dataset. The final results found that the optimised models have high accuracy as well as low latency.

The main conclusions of this thesis are as follows:

- Active learning helps users to write more effective explanations, which improves classifier performance over passively collected explanations.
- Diversity sampling can be the most effective at the early stage of collecting explanations.
- Bayesian active learning is more effective than uncertainty sampling using the predictive probabilities of a standard neural network.

Ethics statement: This project fits within the scope of the blanket ethics application, as reviewed by my supervisor Edwin Simpson.

I have completed the ethics test on Blackboard. My score is 12/12.

1 Project Plan

1.1 Background

The majority of the emphasis on emergency time response systems for social platforms has focused on creating better text classification algorithms to learn from the data. However, obtaining useful annotated datasets can prove difficult [4]. The throughput of general social platform data is unable to accomplish bulk annotation. Therefore, many weakly supervised forms that can achieve the extraction of data with high information value and annotation at a controlled cost have been widely used in classification projects [5, 6]. To reduce the cost of annotation, active learning is used in the backend of social networking sites to accomplish annotation. Active learning is an effective method to solve these problems by selecting unannotated samples with high information content to be annotated by experts [7, 8]. Querying the most informative instances using evaluation algorithms using different sampling is probably the most popular approach in active learning. As a result, query strategies have naturally become a major research hotspot for active learning algorithms, with a variety of optimisation algorithms emerging. Putting annotators into an optimisation loop ultimately achieves high accuracy. At the same time, neural text classifiers are often not comfortable with early uncertainty sampling [9, 10]. They are often considered overconfident in their output of results, and ineffective responses to the complexity of neural networks (NNs) continue to be an important area of research. This is because it has no inherent measure of vulnerability. In contrast, applying the active learning framework not only to the annotation of labels but also to the selection of explanatory features for texts is an innovative experiment, and the model's own difficulties in interpretation can be employed in this active learning approach to increase confidence.

1.2 Motivation

Against the above background, the research focuses on text classification systems' accuracy, interactivity, robustness, and representativeness.

1.2.1 Accuracy

The unifying feature of the social networks represented by Twitter is the sheer volume of data and the amount of noise in the internet language. In particular, when capturing urgent needs, there is a need to classify urgent message categories and dispatch the relevant authorities to solve the issue. However, traditional text classification systems are trained to accomplish high throughput of information. Since information texts have different distributions and the amount of information they represent can greatly affect the model's accuracy, the results of capturing and identifying important emergency information are mostly poor [1]. Therefore, it is important to improve accuracy using semi-supervised models with active learning.

1.2.2 Interactivity

Traditional deep learning models lack the ability to select samples actively. In web-based text with evolutionary capabilities, traditional text classifiers with added explanation tend to be static and therefore struggle with new, rapidly changing samples. Therefore, systems with interactivity mechanisms enable fast labelling of samples and provide accurate interpretation. The performance and accuracy of the model can be further improved by interaction with the user. Especially in emergencies, the interactivity of active learning can make the model more flexible and adaptable to promptly identify and classify public safety-related information, providing critical support for emergency response. The importance of active learning in dealing with interactive scenarios such as the classification of urgent needs on Twitter is therefore self-evident, enabling fast and accurate acquisition of labelled samples through interaction with users and providing strong support for emergency response and public safety.

1.2.3 Robustness

Traditional text classification models often exhibit uncertainty and inadequate generalisation when dealing with text classification tasks. Traditional primary training outputs with explained samples usually connect all instances and are not de-tailored for particular instances. Combining explanation generation with active learning can therefore provide a more robust text classification system that can better address deficiencies in the uncertainty and generalisation capabilities of the model.

1.2.4 Representation

The idea of active learning can assist the model in achieving higher performance with a limited number of labelled samples. By actively selecting samples for labelling, the model can select the most valuable and representative samples in each iteration to improve the model's performance. In addition, the ability to extract rich feature representations and generate explanations in large-scale unlabelled data can reduce a large part of the bias in emergency event multi-classification active learning problems.

1.3 Problem Statement

Firstly, based on the above background, we can conclude that explanation-based neural text classification models need to be trained using a small but informative number of instances and can exploit the maximum potential of the model parameters during training. However, traditional active learning query strategies are based on fixed feature representations, and there is a lack of research on the role of active learning in interpretation generation.

Secondly, how to select instances with high information content to supply to human annotators and detecting the posterior performance of the instances is also the focus of this experiment.

Finally, due to the labour cost, how to simulate the processing process of human annotators is also an effective way to reduce labour.

1.4 Objectives

Therefore, the following research objectives are combined with the above problems:

- Develop a pool-based human-in-the-loop active learning framework for collecting explanations. Integrate extraction algorithms appropriately, design stopping criteria, and investigate whether interactive systems can improve the performance of text classification models.
- Set a baseline(e.g., random data collection) for sampling strategy to pin down the specific sampling strategies.
- Build annotator simulation algorithms(e.g., ChatGPT) and a live annotator simulation program.
- Explore the robustness of the advanced ExpBERT text classification architecture in noisy environments.

1.5 Challenges

- The main challenge of this project is how to improve the training framework of the original ExpBERT textual multiclassification model to improve performance, whether useful information can be extracted through interaction, and whether annotator functionality can be applied not only to label generation but also incidentally to the precision of explanations is a challenge that plays a vital role in the performance of the model.
- It is important to note that most active learning has been applied to binary classification problems, so it is a challenge to apply multi-class active learning to the original framework.
- At the same time, there are multiple query strategies for active learning text classification systems, and the application and comparison of query strategies dramatically affect the performance of the active learning framework. The choice of query strategy needs to be considered in terms of time complexity and user experience.
- Finally, the assessment of the evaluation method, the development of human-in-the-loop stopping criteria. This significantly impacts the framework's performance on the test set and is a noteworthy aspect of this project.

1.6 Thesis Organisation

The overall structure of the thesis based on the research objectives consists of six modules: Introduction, Background, Design, Implementation, Evaluation and Conclusion.

- The introduction phase in Chapter 1 provides an overview of the background to the project, presenting the motivation for the experiments in light of the background, followed by a brief description of the approach to the implementation of the overall interactive system, listing the aims of the research and the challenges faced.

- In Chapter 2, a detailed introduction to the technology of the overall interactive system and its application will be provided, namely the application of the interactive system to a text classification system (Section 2.1), and Section 2.2 provides an introduction to the framework of active learning. Due to the multitude of instance query methods in active learning, section 2.4 provides an introduction to the sampling query algorithm in combination with pseudo-code and formulas, as well as to other active learning frameworks and the stopping criteria needed to end the loop in subsequent sections.

- The overall framework and the design details of the method are described in the first section of the design chapter. Due to the importance of the query strategy, section 3.2 will focus on the query strategy adopted in this report and the design of the evaluation of the representativeness of the examples, and finally, the framework design for explanation generation and training under active learning in ExpBERT.

- The implementation chapter starts with an overview of the experimental environment, introduces the elements of the database and their properties in section 4.2 and then details the implementation process of the simulated user actions in section 4.3. As the model parameters and the sampling strategy are evaluated in the experiments, the design of the evaluation criteria needs to be described in section 4.4. Finally, an overview of the resources required for development is given.

- The chapter on evaluation begins with a description of the selection process of the hyperparameters, the presentation and analysis of the evaluation results for the validation and test sets, and finally completes with the evaluation of the effect of noise.

- In the final chapter, a summary of the previous experiments is presented, followed by an analysis of the overall model's limitations and a discussion of further enhancements and future work.

2 Literature Review

2.1 Representation Engineering with Natural Language Explanations (ExpBERT)

ExpBERT models propose a way to enhance language models' interpretability and knowledge fusion capabilities, focusing on BERT-based models, combining BERT with an explanation generation module, and completing the challenge of language models lacking interpretability and failing to provide predictive explanations explicitly [2]. Figure 1 visually represents how samples with explanations can be combined with a BERT model. First, each tweet is fully concatenated with all explanations to generate a 3-by-3 set. Here the explanations are fixed, i.e. the explanation for each tweet link is not generated based on the tweet; the explanation for the link to the desired tweet is the same. The pre-processed text and explanations are fed into a fine-tuned BERT model, which generates a feature vector for each input sample representing the entire input of length 786. The feature vectors of tweets and explanations are then concatenated to form a size of $768 * E$, where E is the number of explanations, and are used as input data for training and prediction of the classifier model.

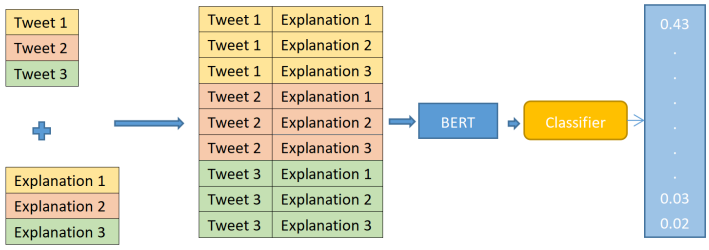

Figure 1: Using BERT to produce representations that form the input to the classifier.

2.2 Human in the loop (HITL)

The traditional natural language processing pipeline is not designed to take advantage of human feedback. However, human in the loop as an essential part of the interactive system, and the simulation of humans in the loop allows for identifying model deficiencies that may not be apparent until real-world testing [11]. Godbole et al. (2004) [12] extended a text classifier based on support vector machine (SVM) active learning to naturally incorporate human input in feature engineering, term inclusion/exclusion, and term and document labelling to make statistically sound decisions. This new type of interaction between humans and machine learning algorithms can be called human-in-the-loop machine learning [13]. Humans in the loop can identify different HITL machine-learning solutions depending on who explicitly controls the learning process [14]:

Active learning (AL) [15]: The system maintains control of the model learning process, with humans acting as a mediator to annotate unlabelled data. However, humans are unable to select unlabelled data based on preferences. AL will be applied as an optimisation framework in this thesis, and the application of active learning is described in detail in section 2.2.
Interactive Machine Learning (IML) [16]: Humans maintain a close interactive relationship with the system. In response to where AL and IML differ shop Dudley and Kristensson (2018) [17] argue that both AL and IML focus on selecting new points for labelling by the user, but in AL, the selection is Since IML is based on AL, they share common disadvantages. However, IML has the added disadvantage of mixing with Human-Computer Interaction techniques (HCI). However, IML has the disadvantage of being combined with Human-Computer Interaction techniques (HCI) and therefore requires unique research [18].
Machine teaching (MT) [19]: Training machine learning models by human teachers. That is, delimiting the knowledge they intend to transfer to the model. Emphasis is placed on the active involvement and guidance of the human teacher in the learning process. Devidze et al. (2020) [20] state that MT is more dependent on teacher expertise than active learning and is less flexible in terms of sample selection and handling complex tasks. Therefore, choosing the appropriate learning method according to the specific needs is crucial.

2.3 Active learning (AL)

As the most popular learning scheme in HITL, active learning systems attempt to overcome the labelling bottleneck by asking questions to unlabelled instances and having them labelled by an expert (e.g. a human annotator) [21]. In short, active learning is the process of identifying the most informative unlabelled to hand over to an annotator, who labels and adds the already-labelled instances to the training process of the model, achieving better performance by using less labelled data. In this project, the instances are large and noisy, the training task is heavy, and the accuracy is low, so active learning is applied to this project to reduce the number of irrelevant instances and increase the accuracy.

2.3.1 AL Process and Scenarios

The active learning process is illustrated in Figure 2, which shows the pool-based active learning process. The machine learning model is initialised and starts by learning the labelled set of instances $\mathcal{L}$, then uses the model to extract features from the unlabelled set of samples and selects unlabelled samples to provide to the human annotator according to a specific selection strategy. The labelled samples are removed from the $\mathcal{U}$ and added to the L-set to update the model. The number of labelled samples is gradually increased by iterative selection, labelling and training. As the performance of the machine learning model improves, it is able to select the most meaningful unlabelled samples for labelling more accurately. Finally, the active learning process is terminated according to

a predefined stopping criterion (e.g. reaching a limit on the number of labelled samples).

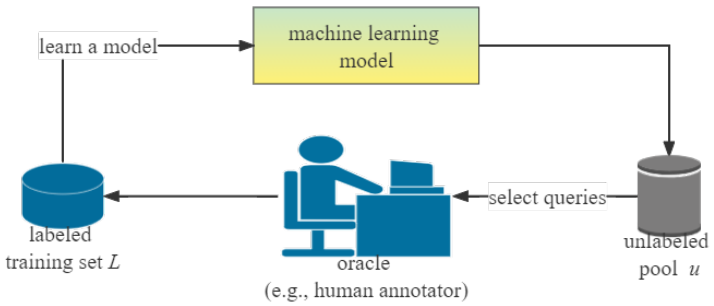

Figure 2: Pool-based active learning process

Depending on the data source, Settles (2009) [15] mentioned three main scenario settings: membership query synthesis [22, 23], stream-based selective sampling [24], and pool-based [25] AL.

Membership query synthesis: queries in this setting are generated by the model, and the machine learning model can request to label any unlabeled instances which are not sampled in the underlying natural distribution [15]. Efficient query synthesis is often practical for more absolute problem domains, such as tasks that employ regression prediction for absolute coordinates, as it allows simple data distributions to be parsed and reasonable data to be constructed for human annotation. However, as seen in Baum and Lang (1992) [26], for complex tasks such as natural language processing, models may produce strings of text that are not easily understood, resulting in humans being unable to make judgements about these confusing texts. In the context of deep active learning, the scenario of member query synthesis can be addressed by generative adversarial networks (GANs) for data augmentation, as GANs are capable of generating instances with a high degree of plausibility [27].

Stream-Based Selective Sampling: This setting can be independent of the input distribution compared to Membership query synthesis. Stream-based sampling draws one unlabelled instance at a time from the actual distribution, and the model decides whether to request the instance's label utilizing an "informativeness metric" or "query policy", which is equivalent to a biased random sampling [28].

Pool-based AL: A common setting for active learning, pool-based AL differs from stream-based sampling in that pool-based AL employs a greedy mechanism to compare the entire dataset before selecting the best query, but the latter receives the data separately for evaluation. However, when the dataset is extensive, selecting the best elements for labelling may become difficult or time-consuming. Therefore, This method is suitable for the more engineering and costly manual labelling task.

2.3.2 Query strategies

This section considers various strategies in active learning, and in the implementation section, the ExpBERT-based textual multiclassification model is used to understand the performance implications.

Random sampling strategies
Random sampling selects instances stochastically, neither based on predictions nor data as well as models, and is therefore used as a baseline for the task. In this case, random sampling is used as the baseline in contrast to the more complex strategies mentioned below, especially when the target pool is too large [29].

Prediction Uncertainty Sampling
Uncertainty sampling, as the name implies, is where the active learner (model) queries the instance that is most difficult to determine its classification. In binary classification problems with probabilistic models, the posterior positive probability of such instances is closest to 0.5 [30]. For more complex multi-label classification problems, however, an entropy-based approach will be used. The more uniform the probability distribution, the higher the entropy and the higher the uncertainty of the random variables, the more informative they are. When probabilities are concentrated in a few data points, this indicates lower uncertainty and less information.

$$x^*_{\mathrm{ENT}} = \arg\max_x - \sum_i P\left(y_i \mid x; \theta\right) \log P\left(y_i \mid x; \theta\right) \tag{1}$$

$P\left(y_i\right)$ denotes the probability distribution at classification i.
In the field of text classification, an alternative approach to the measurement of uncertainty is commonly used, namely, least confident [31]:

$$x^*_{LC} = \arg\min_x P\left(y^* \mid x; \theta\right), y^* = \arg\max_y P(y \mid x; \theta) \tag{2}$$

y^* denotes the most likely class label. This method is equivalent to the entropy-based algorithm valid for binary text classification.
In addition to the two more commonly used measures above, Munro (2020) [13] mentions the margin of confidence, the difference between the two most confident predictions, and the ratio of confidence, which is the ratio of these two predictions.

$$\text{margin\_conf} = 1 - \left(P\left(y^*_1 \mid x\right) - P\left(y^*_2 \mid x\right)\right), \quad \text{ration\_conf} = \frac{P\left(y^*_2 \mid x\right)}{P\left(y^*_1 \mid x\right)} \tag{3}$$

y^*_1 is the most confident, y^*_2 is the second most confident.

Semantic-based Diversity Sampling
Peng, Hao, et al. (2023) [32] proposed a semantic-based diversity sampling approach that can be applied to text classification. The difference with the process of measuring confidence using uncertainty sampling is that the semantic-based

diversity sampling approach uses Euclidean distance to eliminate redundancy in text samples semantically. This ensures that a richer, less repetitive sample is provided to the model (learner) in the subsequent process. This abstraction approach uses the greedy k-centre algorithm of Sener and Saveravarese (2017) [33] for clustering operations. The dataset D contains $n \times m$ unlabelled texts and divides D into n batches, with each sample set containing m instances. It is the result of encoding the dataset. First, select α vectors from Y_i^* to initialise the clusters c_i^0. Here examples will be considered as cluster centres. Then the k-centered algorithm searches u_i^n from $\widetilde{c_i^0}$, which is a set that includes members that are not in Y_i^*. It is the furthest from the centre of all clusters. The algorithm chosen is formulated as follows:

$$\begin{aligned} u_i^0 &= \arg \max_{y_i^k \in \widetilde{c}_i^0} \min_{y_i^j \in c_i^0} \left\| y_i^k - y_i^j \right\|_2^2 \\ u_i^1 &= \arg \max_{y_i^k \in \widetilde{c}_i^1} \min_{y_i^j \in c_i^1} \left\| y_i^k - y_i^j \right\|_2^2 \end{aligned} \tag{4}$$

Where:

$$\begin{aligned} \widetilde{c}_i^0 &= Y_i^* \backslash c_i^0 \\ c_i^1 &= c_i^0 \bigcup u_0 \end{aligned} \tag{5}$$

This is followed by updating the existing clusters to c_i^1 after a loop execution and merging the output into P. All text instances in P converge into a core set that best represents and generalises the dataset D in the semantic space.

Bayesian Active Learning by Disagreement

Bayesian Active Learning by Disagreement (BALD) is a method widely used in active learning for text classification. It is based on Bayesian inference and uncertainty measures and aims to select the most informative samples for annotation. In BALD, a Bayesian neural network or other Bayesian model is first used to demonstrate the text classification task. These models can estimate the probability distribution of each sample belonging to each category and provide a measure of uncertainty about each prediction. Houlsby, N (2011) [34] proposed that the pivotal idea of BALD is to select samples with the highest uncertainty for labelling by making multiple predictions for each sample and calculating the inconsistency between the model's predictions. The learner (model) maximises the uncertainty of the model parameters through the input x. $\mathrm{H}(y \mid x, D)$ represents the uncertainty of the target variable. The second term of the equation represents the uncertainty (entropy) for $\mathrm{H}[y \mid x, \theta]$ under the condition that the parameter θ obeys the posterior probability distribution $p(\theta \mid D)$ of the training dataset D. The average uncertainty is measured by calculating the expected value [35].

$$x^* = \arg\max_x \mathrm{H}(y \mid x, D) - \mathrm{E}_{\theta \sim p(\theta|D)}[\mathrm{H}[y \mid x, \theta]] \mid \tag{6}$$

By calculating the value of BALD, samples with the highest BALD scores can be selected for labelling because they provide the most significant amount of

information while also minimising the uncertainty of the model. Specifically, BALD uses information-gain to measure the uncertainty of each sample, with a higher information-gain indicating a higher information value of the sample. In each active learning iteration, BALD selects the highest information-gain items for annotation and adds them to the labelled training set. The model is then retrained using the updated training set to improve model performance and reduce uncertainty.

2.4 AL loop with Monte Carlo Dropout

In a traditional dropout, training a neural network turns the output of a neuron to zero with a certain probability of reducing overfitting. In MCDO, the network applies dropout to give the network a generative nature [36].

Input Labeled data set L, the unlabeled data U and the untrained classifier f(x).
Output Fully labeled data set L and trained classifier
1:n←Desired data set length
2:q←Query-pool size
3: Q(x)←Query Function
4: T←Number of SFP
5: while L length < n do
6: Retrain f(x) on L
7: P←null
8: for t =0,...,T do
9: insert f(U) into P
10: end for
11: Sort U based on Q(P)
12: Let Oracle assign labels to U-update
13: Insert U-update into L
14: Remove U-update from U
15: end while

Figure 3: MCDO algorithm in AL loop

Multiple outputs are generated for the same input by performing multiple random forward passes. These outputs can be used in various ways to summarise the uncertainty of the model. The pseudo-code for applying the MCDO approach to the active learning loop is shown below. In MCDO, T slightly different models are created by using different dropout samples to approximate Bayesian inference. The query function can use the results of these so-called stochastic forward propagation (SFP) to calculate the uncertainty, as shown in Figure 3. After ranking, the instances with the highest uncertainty are assigned to annotators.

2.5 Small-Text Library

The text classifiers in this project tend to focus on one model and are likely to miss the application of other viable models. However, the time cost of switching models and active learning strategies, as well as the redundancy of the

code, will significantly impact the progress of the experiments. The small-Text library integrates scikit-learn, transformers and PyTorch, and other common libraries that can be applied in a Python environment [37]. The architecture of pool-based active learning for text classification connects the query policy, the classifier and the interface to the abort policy. Not only does it provide a state-of-the-art active learning framework for text classification work, but it also provides a range of classifiers and query policy components to facilitate active learning tasks that can be mixed and matched for rapid application in experiments and applications, making active learning easy to implement in the Python ecosystem. Small-Text offers a more flexible customisation service than the most commonly used ModAL [38] library, where the former is more focused on model integration and the selection of query strategies.

2.6 ChatGPT annotator simulation explanation generation

ChatGPT(Generative Pre-trained Transformer) [39] is a pre-trained language model developed by the OpenAI team in 2018. The basic algorithm is Transformer, a deep neural network structure based on a self-attentive mechanism with robust sequence modelling and representation learning process. Through pre-training as well as fine-tuning, this model can analyse and generate natural-language text and be helpful in multiple scenarios, such as automated answers, intelligent customer support, language translations, etc.

In utilising ChatGPT as an active learning annotator in the ExpBERT model, the main task is to generate suitable interpretations for ExpBERT and provide them to the model to help improve classification performance. Active learning in this configuration generates human-readable interpretations based on the input text by leveraging ChatGPT's powerful generative model [40]. A common approach is to use ChatGPT to create multiple explanations and then select one or more of them as explained instances to pass to the ExpBERT model so that each labelled instance no longer corresponds to the same fully-connected interpretation but to an interpretation tailored explicitly to that instance, which is more relevant.

2.7 Stopping Criteria

Models are commonly trained using active learning using a query strategy similar to the greedy algorithm to select unlabelled instances. As a result, it is easy to overfit the model if the algorithm is carried through to the end, and the model does not stop training when it reaches the desired performance, wasting time and resources. Therefore, a stopping criterion in the loop is needed to prevent this.

Confidence-based Stopping

Vlachos (2008) [41] presents calculating the classifier's confidence using the mean uncertainty on the unlabeled reference set and, for multi-class problems, using SVM classifiers with the SVM margin size as the uncertainty measure. In a loop,

we look for a stopping criterion to find the maximum possible performance of the model and stop the loop. However, Valchos' approach is inappropriate for text multiclassification problems, as the confidence curve stabilises after close to 500 iterations but does not show a peak due to the instability of artificial intelligence. Therefore, the peak confidence criterion based on the mean reference uncertainty is not applicable to this model.

Gradient-based Stopping

Gradient-based Stopping addresses the drawback that Confidence-based Stopping cannot use peaks as a stopping strategy. It combines performance and uncertainty convergence stopping criteria and determines whether the active learning process should be stopped by observing the change in gradient. Precisely, the angle is calculated using the following formula:

$$g = (\text{median}(w2) - \text{median}(w1))/1 \tag{7}$$

In this case, $w1$ represents the previous last n values, and $w2$ represents the last n values. Meanwhile, a window of size k = 100 produces good results in noise mitigation while still responding fast enough to changes in the gradient [42]. Thus, the AL process is terminated when the current deterministic or estimated performance is a new maximum, whilst g is positive and falls beneath a predefined level.

References

[1] R. McCreadie, C. Buntain, and I. Soboroff, "Trec incident streams: Finding actionable information on social media," 2019.

[2] S. Murty, P. W. Koh, and P. Liang, "Expbert: Representation engineering with natural language explanations," *arXiv preprint arXiv:2005.01932*, 2020.

[3] Y. Baram, R. E. Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, no. Mar, pp. 255–291, 2004.

[4] S. Prabhu, M. Mohamed, and H. Misra, "Multi-class text classification using bert-based active learning," *arXiv preprint arXiv:2104.14289*, 2021.

[5] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 601–614, 2011.

[6] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 606–617, 2011.

[7] B. Settles, "Active learning literature survey," 2009.

[8] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and information systems*, vol. 35, pp. 249–283, 2013.

[9] C. Schröder and A. Niekler, "A survey of active learning for text classification using deep neural networks," *arXiv preprint arXiv:2008.07267*, 2020.

[10] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[11] J. E. Tomaszewski, "Overview of the role of artificial intelligence in pathology: the computer as a pathology digital assistant," in *Artificial intelligence and deep learning in pathology*. Elsevier, 2021, pp. 237–262.

[12] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti, "Document classification through interactive supervision of document and term labels," in *Knowledge Discovery in Databases: PKDD 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings 8*. Springer, 2004, pp. 185–196.

[13] R. M. Monarch, *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.

[14] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, "Human-in-the-loop machine learning: A state of the art," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3005–3054, 2023.

[15] B. Settles, "Active learning literature survey," 2009.

[16] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *Ai Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[17] J. J. Dudley and P. O. Kristensson, "A review of user interface design for interactive machine learning," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 8, no. 2, pp. 1–37, 2018.

[18] C. J. Michael, D. Acklin, and J. Scheuerman, "On interactive machine learning and the potential of cognitive feedback," *arXiv preprint arXiv:2003.10365*, 2020.

[19] G. Ramos, C. Meek, P. Simard, J. Suh, and S. Ghorashi, "Interactive machine teaching: a human-centered approach to building machine-learned models," *Human–Computer Interaction*, vol. 35, no. 5-6, pp. 413–451, 2020.

[20] R. Devidze, F. Mansouri, L. Haug, Y. Chen, and A. Singla, "Understanding the power and limitations of teaching with imperfect knowledge," *arXiv preprint arXiv:2003.09712*, 2020.

[21] R. Caceffo, G. Gama, and R. Azevedo, "Exploring active learning approaches to computer science classes," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, pp. 922–927.

[22] D. Angluin, "Queries and concept learning," *Machine learning*, vol. 2, pp. 319–342, 1988.

[23] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, "Functional genomic hypothesis generation and experimentation by a robot scientist," *Nature*, vol. 427, no. 6971, pp. 247–252, 2004.

[24] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 150–157.

[25] D. D. Lewis, "A sequential algorithm for training text classifiers: Corrigendum and additional data," in *Acm Sigir Forum*, vol. 29, no. 2. ACM New York, NY, USA, 1995, pp. 13–19.

[26] E. B. Baum and K. Lang, "Query learning can work poorly when a human oracle is used," in *International joint conference on neural networks*, vol. 8. Beijing China, 1992, p. 8.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[28] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 150–157.

[29] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[30] A. Raj and F. Bach, "Convergence of uncertainty sampling for active learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 310–18 331.

[31] R. Hu, B. Mac Namee, and S. J. Delany, "Active learning for text classification with reusability," *Expert systems with applications*, vol. 45, pp. 438–449, 2016.

[32] H. Peng, S. Guo, D. Zhao, Y. Wu, J. Han, Z. Wang, S. Ji, and M. Zhong, "Query-efficient model extraction for text classification model in a hard label setting," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 4, pp. 10–20, 2023.

[33] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[34] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[35] D. J. MacKay, "Information-based objective functions for active data selection," *Neural computation*, vol. 4, no. 4, pp. 590–604, 1992.

[36] E. Tsymbalov, M. Panov, and A. Shapeev, "Dropout-based active learning for regression," in *Analysis of Images, Social Networks and Texts: 7th International Conference, AIST 2018, Moscow, Russia, July 5–7, 2018, Revised Selected Papers 7*. Springer, 2018, pp. 247–258.

[37] C. Schröder, L. Müller, A. Niekler, and M. Potthast, "Small-text: Active learning for text classification in python," *arXiv preprint arXiv:2107.10314*, 2021.

[38] T. Danka and P. Horvath, "modal: A modular active learning framework for python," *arXiv preprint arXiv:1805.00979*, 2018.

[39] A. Azaria, "Chatgpt usage and limitations," 2022.

[40] Y. Shi, H. Ma, W. Zhong, G. Mai, X. Li, T. Liu, and J. Huang, "Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs," *arXiv preprint arXiv:2305.03513*, 2023.

[41] A. Vlachos, "A stopping criterion for active learning," *Computer Speech & Language*, vol. 22, no. 3, pp. 295–312, 2008.

[42] F. Laws and H. Schütze, "Stopping criteria for active learning of named entity recognition," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 465–472.

A Project Timeline

- Week 1(Jun 26 - Jul 9): I will complete the human-in-the-loop framework using active learning.
- Week 2(Jul 10 - Jul 16): I will simulate the users(annotators) to give explanations to unlabeled instances using a random sampling strategy.
- Week 3(Jul 17 - Jul 23): I will use different sampling strategies of active learning and adapt the method to the explanation generation process with a text classifier.
- Week 4(Jul 24 - Jul 30): I will use more advanced deep active learning methods as a sampling strategy.
- Week 5(Jul 31 - Aug 6): I will use some standard defaults for BERT classifiers and evaluation metrics for evaluation to run the experiment.
- Week 6(Aug 7 - Aug 13): The first version of the code will be complete, and I will carry out the code review and optimize the algorithms; At the same time, I will start writing the Chapter on design.
- Week 7(Aug 14 - Aug 20): After two iterations of the code review, I will execute the final evaluation of the code and peer review; At the same time, the Chapter of Implementation part of my thesis will be written.
- Week 8(Aug 21 - Aug 27): The Chapter on Evaluation and Conclusion will be written.
- Week 9(Aug 28 - Aug 31): Thesis review and peer review.
- Week 10(Aug 31 - Sep 5): Oral presentation will be prepared and recorded.

Task Visualisation:

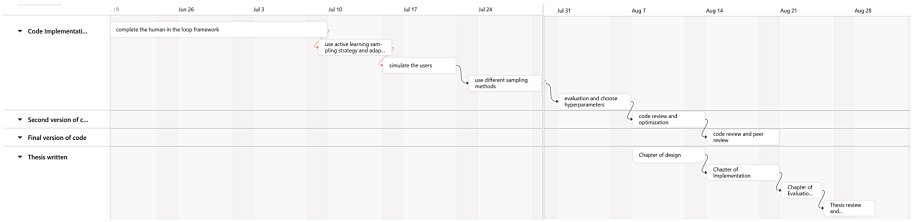

Figure 4: Task Timeline

B Risk Assessment

Table 1 shows the risks and mitigation.

Table 1: Risks and Mitigation

Risk	Likelihood	Severity	Mitigation
Computer problems (stolen or broken)	Medium	High	Back up promptly and update the progress on GitHub.
The GitHub server breaks down	Low	High	Back up code on different version control platforms.
Body issues	Medium	Medium	Take care of body and finish the task as soon as possible.
Performance of the model not ideal	High	High	Ensure the experiments are carried out correctly and try to identify the reasons why the method did not work smoothly.