DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

A Comparison Between the Use of Unlabeled and Weakly Labeled Data in Active Learning for a Text Classification Problem

Michal Cizevskij

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

A Comparison Between the Use of Unlabeled and Weakly Labeled Data in Active Learning for a Text Classification Problem

Vergleich des Ansatzes von ungekennzeichneten und schwach gekennzeichneten Daten bei der Verwendung von aktivem Lernen für ein Textklassifikationsproblem

Author:	Michal Cizevskij
Supervisor:	Martin Bichler
Advisor:	Fabian Pieroth, Fabian Hertwig
Submission Date:	15.06.2023

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.06.2023 Michal Cizevskij

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Fabian Pieroth and Fabian Hertwig, for their invaluable guidance and unwavering support throughout this research. Their expertise and mentorship have been instrumental in the successful completion of this work. I am also deeply thankful to my family for their unwavering support, love, and encouragement. Their presence and belief in me have been a constant source of motivation and strength.

Abstract

The rising prominence of Data Centric AI has brought into focus key techniques such as Active Learning. This thesis introduces two methodologies that aim to leverage the knowledge of large language models to minimize labeling costs. The first method, termed 'Active Learning Plus', is a novel blend of Active Learning, Pseudo Labelling, and weak label predictions from large models. The second approach, called Active Learning Initial, involves "warming up" the model on weak predictions, followed by standard Active Learning iterations. These innovative approaches are designed to enhance the efficiency of the (active) machine learning process. The experimental outcomes, obtained on the AG News data set, indicate that both proposed methods can be effective and outperform the standard Active Learning approach and can help to achieve a benchmark with fewer annotation costs, given a weakly labeled data set is available.

Contents

1 Introduction

Since the past few years, the use of Artificial Intelligence is significantly gaining in importance [14]. With the introduction of ChatGPT-3 to the public [3], Artificial Intelligence has evolved from a buzzword to an integral part of people's daily lives [18].
Even though Artificial Intelligence is exponentially improving. We find ourselves at the stage where AI combined with human interaction is at its best, often referred to as Collaborative AI [41], leverages the strengths of both human expertise and AI capabilities to achieve optimal outcomes. To further evolve AI and lessen the necessity for human interaction. A vast amount of data is required to fuel the training of the machine models. Although one could think that in today's world, there is an abundance of data, it is essential to differentiate between clean and noisy data.
This is where the notion of Data Centric AI takes effect. Generally, the goal of Data Centric AI is to shift the focus from optimizing the algorithmic elements of Machine Learning to improving the data the machines are trained on, according to an emerging studies by Zha et al. [46] and by Jarrahi et .al [15]. Adopting a data centric approach is of utmost importance, especially considering that even established benchmark data sets, often used in academic research, have been proven to contain errors. A selection of ten data sets, including CIFAR-10, WikiText, MNIST, among others, were studied. These data sets were found to be erroneous, with an estimated average error rate of at least 3.3% according to Northcutt et al. [28]. This underlines the significance of data accuracy and integrity in the field of machine learning research.
The Term Data Centric AI encompasses any methodology that improves a given data set. Mainly it trickles down to these big tasks: labelling data, augmenting data, diversifying data, correcting data and other general data processing techniques which help with the training efficiency [25].
In this Thesis, the main focus is on a technique for data labelling and model training called Active Learning. The goal of Active Learning (AL) is to minimize the labelling effort of an unlabelled data set while retaining a high model accuracy and reducing training time.
This research compares the performance of the aforementioned technique on unlabeled and weakly labeled data. Weakly labeled data in this context is data that is deemed unreliable and can contain errors e.g wrong labels. The weakly data is obtained with the help of a larger models predictions. The large model has been pre-trained for

another but related task. This approach can also be considered as a form of Transfer Learning. In contrast, strongly labeled data refers to data for which we have high confidence in the accuracy of the labels, are obtained through expert annotation. The purpose of this research is to determine whether it is possible to spare resources in the training process of language model by utilizing the a larger language model in combination with Active Learning, in comparison to utilizing Active Learning alone.
Due to time and cost constraints a data set with known ground truth has been used so that the human annotation can be simulated programmatically. Additionally a simple 'Large Model' is used and later on the weakly labelling is simulated with a fixed error rate for the data labels. This research focuses on one large but simple natural language data set AG NEWS that is balanced and classified into 4 classes [48]. For faster training process a transfer model BERT base cased is used [6].
This thesis makes a contribution to the field of applied active learning practices, primarily focusing on the minimization of annotation costs. It provides potential solutions to the question of handling textual data sets that have unreliable labels. The study evaluates whether it is more beneficial to leverage this data set for knowledge or whether it's more prudent to discard these unreliable labels and apply active learning directly. The same consideration applies to the scenario of using a larger, pre-trained language model to generate these pseudo labels. The proposed approaches work in a similar fashion to the studies "Cost-Effective Active Learning for Deep Image Classification" by Wang et al. (2017) [38] and "Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification" by Wu et al. (2018). With the distinctive aspects that these pseudo labels are derived from the predictions or clustering results generated by the large language model. Moreover, this research explores the determining factors for these decisions, aiming to identify the tipping point at which one choice becomes more beneficial than the other.
It begins with an introductory section that offers an overview of the thesis. The subsequent chapter on Related Work discusses relevant research papers to this study. Moving forward, the Background section examines the specific technologies and methodologies employed in the research. Thereafter, in the Methodology chapter, the proposed framework and its implementation are thoroughly analyzed. The Results section presents the findings and outcomes, showcasing the obtained results. Following this, the Discussion section provides an in-depth analysis and interpretation of the results. Finally, the Conclusion offers a summary and synthesis of the research study, providing a conclusive summary of the findings.

2 Related Work

In recent years, the perspective on machine learning applications has shifted, with an increasing focus on viewing these as data-centric problems rather than strictly algorithmic ones. This pivot towards a data-centric approach is predominantly due to the realization that data quality significantly impacts the performance of machine learning models. For instance, acquiring labeled data in certain domains can be a strenuous and costly process.
This is often the case for data that originated from the medical domain. Specifically, the task of annotating textual data has been proven to be difficult given its subjective and context-dependent nature. As a result, a significant portion of time and costs is spent on data preparation.
This literature review addresses various strategies for efficient data processing and model training, with the aim to reduce the amount of labeled data required for effective model training. The primary focus is on Active Learning, Pseudo Labelling, and their combination.

2.1 Active Learning

To provide a general overview of Active Learning, it is helpful to reference two main sources: A web survey [33] conducted by Burr settles that introduced and summarized uses of Active Learning. Even though the survey was published in the year of 2009, it grants a great general overview of AL. Additionally, it helps to thematically classify AL in the domain of Machine Learning through giving the necessary context and examples. The second contextual source is the book by Robert Munro, Human-In-The-Loop. Here, the author gave his expertise and pointers from the experience he has gained personally while developing active learning systems. The book covers the key methods for developing Active Learning systems, including the most important approaches such as uncertainty sampling and diversity sampling [27].
Numerous studies have shown the effectiveness of Active Learning compared to passive Learning [47]. Active Learning has been successfully applied across multiple domains, including computer vision [1], natural language processing [45], and bio-informatics [26], among others.
The focus of Active learning is to attain cost-effectiveness through maximization of

the learning effect from a limited number of labeled data instances [32]. This is accomplished by using sampling techniques, where only the most informative data is chosen for model training. Consequently, the necessary training data is kept to a minimum. One of the early references to Active Learning dates back to 1994 in the work of Lewis et al. [20].
In this paper, the authors introduced the fundamental concept of uncertainty sampling [20], where the program queries instances with the help of the model's predictions to be labeled by the oracle. Uncertainty sampling aims to select instances that the model is least certain about. Commonly employed methods to assess uncertainty in active learning include entropy, margin of confidence, ratio of confidence, and least confidence [27].
Diversity sampling is another type of sampling strategy that seeks to counteract model overfitting to the training set, a common issue when models are applied to real-world scenarios. Compared to uncertainty sampling, however, diversity sampling has been less commonly used. It aims to provide the model with a diverse sample to enhance its overall understanding of the data set. One potential risk with diversity sampling is its tendency to focus solely on the outliers of the data set, which can potentially skew the model's predictions [27].
There is a variety of scenarios for applying active learning, encompassing different techniques and strategies based on the specific requirements and constraints of the data and task at hand. For particularly small datasets, Membership Query Synthesis can be beneficial. In this approach, the Learner also takes on the task of synthetically generating augmented data. These instances can be a part of a given datum, a combination of multiple data, or just regular data instances [39]. However, in real-world applications, problems can occur when using membership query synthesis. The learner might generate a datum that is unrecognizable for the oracle (e.g., human annotator), leading to practical difficulties [34, 39].
The Pool-Based Scenario is a common strategy where the confidences of the model are calculated on the whole dataset and later evaluated [36]. In this approach, the data resides inside a large pool, and each iteration queries the whole pool for the next annotation sample.
Batch Mode Active Learning can be seen as an extension to the Pool-based scenario. The primary difference being that instead of querying only one sample each AL iteration, the Learner queries multiple data instances simultaneously [12, 11]. This mode can enhance efficiency, especially when dealing with large datasets such as AG NEWS [48], where individual instance selection would be prohibitively time-consuming. This proves particularly advantageous in scenarios with limited computational resources.
However, it is important to mention that Active Learning is prone to bias. According to Farquhar et al. (2021) and B. Settles (2011), the training data may deviate from the

population distribution, leading to potential biases in the learning process [7, 34]. Thus, some authors recommended combining sampling techniques such as diversity and uncertainty sampling [27]. But handling Active Learning bias is not the focus of this thesis.

2.2 Transfer Learning and BERT

Transfer learning is the concept of transferring knowledge from one problem domain to another with the help of machine learning [40]. A common use case is taking a model that has been trained on a large corpus of data and then fine-tuning it on a specific use case. An exemplar of such a strategy is BERT (Bidirectional Encoder Representations from Transformers) [6], a technique widely used for various natural language processing tasks. As demonstrated in the paper "Comparing BERT against traditional machine learning text classification," BERT has shown superior performance over other traditional approaches, especially in the field of text classification [8].
Moreover, the paper "Active Learning by Acquiring Contrastive Example" by Margatina et al. (2021) [23] proposed its own sampling method for Active Learning with BERT for a classification task on the AG NEWS data set. But instead of using Multi-class classification, the authors set the category 'World' as positive labels and the other three categories as negative labels. It's important to note that this paper has not undergone peer review and is cited here solely for its application of these techniques in combination. Despite this, the study underscores the effectiveness of BERT for text classification tasks, especially when used in tandem with Active Learning.

2.3 Pseudo Labelling

Pseudo Labelling is an alternative form of semi-supervised learning. This method is typically employed when the availability of labeled training data is limited, and there is a desire to improve the model's performance. Furthermore, Pseudo Labelling serves as a form of regularization, helping to prevent model overfitting. Either an external pre-trained model is used to make predictions on the unlabeled data (which can also be considered a form of transfer learning), or the model is first pre-trained on the limited available data and then used to make predictions on the unlabeled data. For predictions with high confidence, the labels are accepted as fixed (pseudo) labels and combined with the certainly labeled instances for training.
In this paper, Lee and others (2013) showed that Pseudo Labelling is a viable semi-supervised learning technique, outperforming other approaches on the MNIST dataset even with a small amount of labeled data [19]. In their 2021 publication, Rizve and

colleagues established Pseudo Labelling as a potent regularization technique, especially in managing model uncertainty. They demonstrated that this method can achieve results comparable to those of consistency regularization-based Semi-Supervised Learning (SSL) approaches [30]. In conclusion, Pseudo Labelling proves beneficial not only in scenarios with limited data but also enhances the model's generalization capabilities.

2.4 Combination of Active Learning and Pseudo Labelling in Literature

After reviewing these methods profoundly, the question arises as to whether these two techniques can be combined to further reduce the necessary amount of labeled data. The paper titled "Cost-Effective Active Learning for Deep Image Classification" (Wang et al., 2017) combined the aforementioned approaches to create a cost-effective framework with a small labeled data set and a larger unlabeled set [38].
The framework begins by labeling an initial set of data and training a Convolutional Neural Network. In each iteration, the model is utilized to make predictions on the unlabeled data. Low-confidence samples are selected and labeled, combining them with high-confidence samples. This combined set is then used for training, repeating the process until satisfactory results are obtained. Contrary to the aforementioned study, the research presented here focuses on textual data rather than image data. Moreover, the dataset considered in this research is comparatively simpler. An interesting aspect to note from the previous study is the use of a dynamic threshold for pseudo label acceptance [38]. This approach can adjust itself according to the learning capability of the model, but it also introduces an additional element of variability into the process.
In a subsequent study, "Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification" (Hao Wu and Saurabh Prasad, 2018), the authors managed to combine the two approaches of Pseudo Labelling and Active Learning as well [43]. However, their approach differs in several key aspects. Initially, the authors clustered their entire data set, which comprises both labeled and unlabeled data. They then pre-trained their model using these pseudo labels. Subsequently, they fine-tuned the model by discarding the final layer and training it on the accurately labeled data. Similar to the aforementioned method, the authors demonstrated that their approach outperforms established state-of-the-art techniques. It's crucial to note that their work is based on a particularly challenging classification problem concerning hyperspectral image data.
Overall, the integration of Pseudo Labelling and Active Learning presents a compelling synergy in the field of cost-effective machine learning. The literature review highlights the significant impact of combining Active Learning and Pseudo Labelling on reducing

the labeled data required for training. Moreover, an intriguing and unexplored question arises: What is the threshold at which additional knowledge from a pre-trained model can be effectively leveraged when generating Pseudo Labels for training an Active Learning-based model? The threshold, in this context, pertains to an evaluation metric such as accuracy, recall, precision, or a similar measure. Although previous research predominantly focuses on the domain of Computer Vision, I have chosen to conduct my study in the domain of Natural Language Processing.

3 Background

3.1 Active Learning

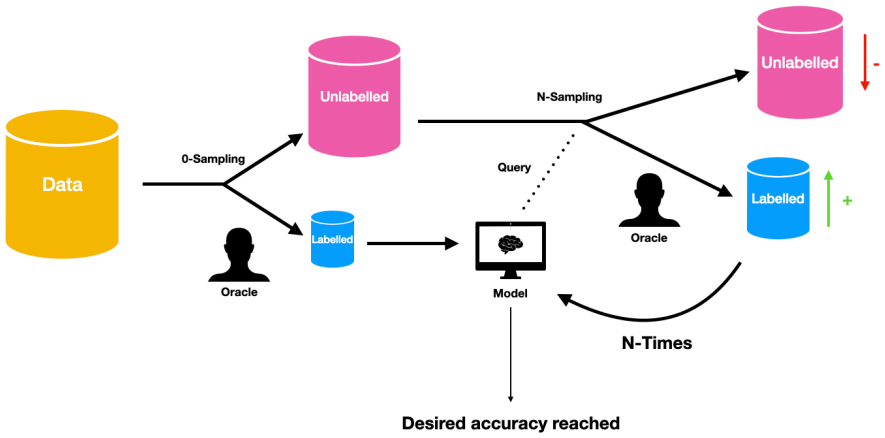


Figure 3.1: Active Learning Loop.

This section clarifies key concepts relevant to this thesis. Active Learning is an iterative process that requires the supervision by an entity often referred to as the oracle in the training (learning) stage of a model. The oracle is either a human or another trustworthy information source. Therefore AL is often categorized as a Human-in-the-Loop process. The role of the oracle is to label data points when queried by the system. These data points can be chosen with the help of various sampling methods. The goal is to select

the most informative data instances. Subsequently these data points are used for model training.
Generally Active Learning is used on data where labelled instances are difficult to obtain or manual annotation is necessary. Therefore, one will often see AL utilised in fields where expert knowledge is required, such as Medicine [10] [44]. Furthermore Active Learning can be categorized by the used sampling method. Most commonly used are: uncertainty sampling, density-based sampling, query-by-committee, diversity sampling and the default random sampling. All the sampling methods decide which unlabelled data should be next to be annotated by the oracle. For additional clarification, refer to Figure 3.1.

3.1.1 Sampling Techniques

- Uncertainty sampling is a collection of techniques where the data is chosen by the uncertainty of the current model on the unlabelled data set.
- Where as density sampling is a technique where the next data points are chosen by their vicinity to the decision boundary. First one has to utilize a Clustering algorithm on unlabelled data. Then one can identify the densest parts for sampling.
- The query-by-committee is a sampling method where multiple models are used. Each iteration the committee collectively decides on the next data point that should be labelled and later on the whole committee is trained on the newly labelled instances [35].
- Diversity sampling chooses its data points based on the nature of the data set. The goal is to find instances that vary from the labelled subset. The purpose is to achieve the highest information gain from the newly sampled data.
- The random sampling selects the next data points randomly.

In this Thesis the main focus will be set on the uncertainty sampling techniques. Therefore, they will be explained in detail in the following subsection.

3.1.2 Uncertainty Sampling

This section explores the most common Uncertainty Sampling techniques, which play a central role in this research. These techniques are utilized to calculate the uncertainty of the model's predictions on the unlabelled data set. Initially, the model generates logits for the unlabelled instances. To convert these logits into a probability distribution

over multiple classes, an activation function like $softmax(x\_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$ is applied. Subsequently, one of the following methods is applied. Each returns a value in the interval [0, 1], where 1 indicates the most uncertainty and 0 denotes the highest confidence.

Entropy

$$-H(x) = \frac{-\sum_{x \in X} p(x) \log_2 p(x)}{\log_2(K)} \tag{3.1}$$

In the equation the variable K represents the total amount of classes in classification problem. The numerator in this equation corresponds to Shannon's Information entropy, which measures the uncertainty or disorder of the predicted probabilities for each class. The denominator serves the purpose of normalization, ensuring that the entropy value falls within the range [0, 1] regardless of the number of classes [27] [5].

Least Confidence

$$LC(x) = 1 - \frac{\max(x)}{\sum(x)} \tag{3.2}$$

Here the highest confidence value is deducted from the highest possible confidence to account for uncertainty [4] [27].

Margin of Confidence

$$MoC(x) = 1 - \max(x) - \max_{i \neq argmax(x)}(x_i) \tag{3.3}$$

In this approach, the uncertainty value is calculated by taking the difference between the highest probability and the second-largest probability [31] [27].

Ratio of Confidence

$$RoC(x) = \frac{\max(x)}{\sum(x)} \tag{3.4}$$

The RoC method computes a ratio instead of a difference, similar to the previously mentioned approach. This ratio offers insight into the relative level of confidence or certainty associated with the maximum score in and the second highest confidence [27]. For Further reading refer to [33] and [27].

3.2 Pseudo Labelling

Pseudo Labeling is a technique predominantly employed in the domain of semi-supervised learning. This approach is typically invoked when the aim is to enlarge the training data set size at a minimal cost, particularly when the available annotated data is scarce and the labeling process is is exceptionally difficult or expensive to label. Other reasons for its usage include its function as a form of regularization and its capacity to further draw knowledge from unlabelled data. The procedure of Pseudo Labeling can essentially be distilled into three fundamental steps:

1. (a) Utilize a clustering technique from the unsupervised field and accept the clusters as labels.
   (b) Alternatively, pre-train the underlying model on the limited available data and make predictions on the unlabelled data.
2. (a) Assign pseudo labels to instances within each cluster.
   (b) Or assign predicted labels to instances that exceed a specific confidence threshold.
3. Train the model on the combined set of pseudo-labelled and correctly labelled instances.

3.3 BERT

The acronym BERT stands for Bidirectional Encoder Representations from Transformers, is a transformative model in Natural Language Processing (NLP). Introduced by Google in 2018, it introduced innovative approaches to language modeling techniques. An intriguing characteristic of BERT is its pre-training approach. Unlike many traditional models, it is pretrained on a voluminous unlabelled text corpus, which helps it to learn the context of words and their co-occurrence. Furthermore, fine-tuning is performed on the final layer of the model, allowing for its application to a wide range of tasks.
In its acronym, "Bidirectional" stands for the new approach in processing textual data in a bidirectional sequential fashion, BERT evaluates context from both left and right. This ability to comprehend context from two directions allows for a more robust understanding of the semantic contents of text tasks.
The "Encoder" component of BERT concerns the conversion of text data into a machine-readable format. A fundamental tool in this process is the BERT tokenizer. Initially it fragments the text on fundamental delimiters such as commas and whitespace. After that it is tasked with breaking down the text data into subwords and special tokens. In

the next step the tokenizer assigns specific Token IDs to the subwords and attention masks, which helps the model distinguish between meaningful instances and those that can be overlooked.tasks.
Lastly, "Transformers" form the foundation of the BERT model. These NLP models operate based on attention mechanisms and were first introduced by Vaswani et al. in the paper "Attention is All You Need" [37]. For a more elaborate explanation please refer to the article "Pre-training of Deep Bidirectional Transformers for Language Understanding" by Jacob Devlin et al. [6]. This seminal work offers a comprehensive discussion of the theories and methodologies that lay groundwork for BERT.

4 Methodology

Within this methodology section, the specific manner in which the research has been conducted will be described, namely how the frame work that was implemented to research the question:
"Is it possible to minimize the annotation workload by combining a weak annotator with active learning, rather than exclusively utilizing active learning?". To see the implementation for yourself please refer to this public GitHub repository: `https://github.com/Mscix/BA`. Initially a short overview of the research environment in which the study operates is provided. Afterwards this section begins with a concise description of each important component class. In the next phase, the general worklfow of the program is outlined. This is followed by a discussion on the data set and its preparation. Subsequently, the focus shifts to the model and its training process. Then the different modes/approaches and that are covered by this implementation are considered. Finally, the experimental set up is described, along with a elaboration on the evaluation metrics that are used to measure the effectiveness of the proposed methods. The developed framework was implemented in Python 3.9, with Pytorch as the underlying Machine Learning framework. Initial debugging was conducted locally on a CPU. However, for extensive evaluation, resources from Paperspace Gradient were employed, where computations were performed on a Compute Unified Device Architecture (CUDA) device with the support of a GPU RTX4000. CUDA is a parallel computing platform and application programming interface model created by NVIDIA that utilizes its GPUs to accelerate computing tasks.

4.1 Project Overview

The project adheres to the Singleton structure, where the Main class serves as the main controller and code entry point. The general workflow of the application during a single iteration is illustrated in the figure below (4.1).

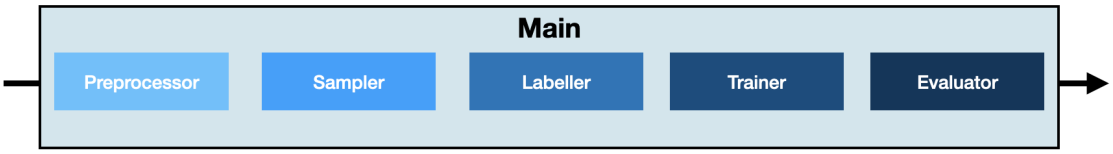


Figure 4.1: Data Flow.

In the `Preprocessor` the chosen subset for example `"big.csv"` that corresponds to 4000 data points is then divided into 80% training set and 20% validation set and subsequently shuffled.
`Preprocessor` holds the data separately and generally is responsible for data management. The training set is then divided into 2 disjoint sets: labeled data and and partial data. The name partial is the term for either unlabeled data or data that has been weakly labeled.
Most important is the `to_dataloader` method that is responsible for the transformation of the underlying pandas data-frame to an object `DataLoader` that can be fed into the model. This method is invoked each time a prediction, training, or evaluation of the model is performed. Generally `DataLoader` is used for faster batched data processing.

The `Sampler's` responsibility is to sample data from the partial data set. The specific method choice is made at runtime. The `Sampler` uses predictions from the model-in-training and a specific sampling method to pick data points that should be labeled next, conforming to the approach of Batch Mode Active Learning. The uncertainty values are computed for the subset of data. Subsequently, these values are sorted and a portion equivalent to the sampling size is passed to the labeler for further processing.
Additionally when using the AL+ approach one can choose to add pseudo labels to the training set. These labels are chosen based on the given delta that is the cut off confidence for the data points to be included in the training's set. The user can also opt to use the generated weakly labels or accept the predicted class by the model-in-training.
For the sampling techniques the main uncertainty sampling techniques and random sampling are implemented: Entropy, Least Confidence, Margin of Confidence and Ratio of Confidence.
The `Strong Labeler` is a concept that replaces human annotation. This class is responsible for labeling data that a human would have to. More specifically it labels data that was sampled during uncertainty sampling or random sampling. The labels that are set by this class are considered ground truth and are looked up in the control data set.
The `Weakly Labeler` acts as a simulated large language model that is used to leverage the available knowledge and to reduce the labeling effort. It pre-labels the unlabelled set.

These labels can be then considered pseudo labels and additionally used during training. The `Weakly Labeler` can either be the K-Means algorithm, which computes distances between word embeddings generated by the ChatGPT model "text-embedding-ada-002" [3], or the `Custom Labeler`, a simulated Weakly Labeler that allows for a custom labelling error rate for pseudo labels. It is essential to note the process by which the `Custom Labeler` labels the given data. The `Custom Labeler` takes a correctly labeled control data as input, then randomly samples a portion of the data set equivalent to the passed error rate. For each instance, it verifies the "Class Index" value and then randomly reassigns it to one of the three other possible values. For instance, if the label value at index 3089 is "1", it will then be re-labeled to either "0", "2", or "3" with an equal probability of $\frac{1}{3}$ for each.
The class `Trainer`, trains the model using provided data, optimizing its performance through specific techniques and algorithms.
The class `Evaluator`, assesses the trained model's performance using various evaluation techniques. The following table gives an overview of the arguments that can be passed to the program. The program is highly configurable so that a variety of experiments can be run. Most of the decisions are made at run time meaning it is highly dependent on the users command line input.

Argument	Description	Input Type
`-p`	Path to the training set, must be in CSV format	String
`-tp`	Path to the test set, must be in CSV format	String
`-m`	Program mode: AL+ (Active Learning Plus) , AL (Active Learning), Standard	String
`-sm`	Sampling method: Random, Least Confidence (LC), Entropy (EC), MC (Margin of Confidence), RC (Ratio of Confidence)	String
`-d`	Confidence delta, determines the pseudo-label acceptance threshold	Float, $x \in [0, 1]$
`-ait`	The number of active learning iterations to perform	Integer
`-ns`	Number of instances to select in each active learning iteration	Integer or Float
`-iss`	Number of instances to select for the initial step	Integer or Float
`-err`	The error rate with which the Custom Labeler will label the training data	Float, $x \in [0, 1]$
`-r`	This argument determines whether the model is reset each active learning (AL) iteration	Boolean
`-pat`	This is the tolerance parameter for Early stopping	Integer ≥ 0

Table 4.1: Summary of command line arguments

In the general workflow of the program, the first step involves initialization where main components are set up based on the passed parameters. This initiates the active learning loop. In the initial iteration, a random sample is taken which is strongly labeled and added to the training set.

In case the mode is Active Learning Initiatl (ALI) or Active Learning Plus (AL+), there is a phase of weak labeling. In the case of the mode ALI, the system will train the model on a combined set of strongly and weakly labeled data. Otherwise the model will be trained only on the strong labeled train set. Following this, the first round of evaluation takes place.

From the second iteration onwards, the process becomes slightly different. The active learning loop uses uncertainty sampling (although it can continue with random sampling if desired), which makes use of the model's prediction capabilities. The sampling phase returns three key data slices: a sample that will be strongly labelled, the remaining instances of the partial set, and the pseudo labels that can be used for future training.

Next, depending on whether the system is in AL+ mode, it either combines both pseudo labels and strong labels for training, or uses only strong labels. It is important to note that pseudo labels are linked to instances classified with lower confidence, while strong labels are associated with instances classified more confidently, such as those annotated by humans.

Finally, the model undergoes another round of training using this data, followed by an evaluation step to assess the model's performance. This iterative process continues for as many cycles as specified, allowing the model to progressively learn and improve its performance. Below one can see the pseudo code for the main loop.

Algorithm 1 Active Learning Plus Framework

```
procedure Active Learning Main Loop
    Initialize main components depending on passed parameters
    Begin Active Learning Loop
    Initial Iteration:
        partial = []                          ▷ Either unlabelled data or weakly labelled data
        conf_delta = d                        ▷ Command line argument, default = 0
        sampled, partial, _ = sample(data, random_sampling, _)
        labelled_data = strong_labeler.label(sampled)
        if Mode == ALI then
            partial = weakly_labeler.label(partial)
            train_set = concat(partial, labelled_data)
        else if Mode == AL+ then
            partial = weakly_labeler(partial)
            train_set = labelled_data
        else then
            train_set = labelled_data
        end if
        model = train(model, train_set)
        evaluate(model, test_set)
    for each i-th AL-Iteration do
            sampled, partial, pseudo_labels = sample(partial, uncertainty_sampling, conf_delta)
            labelled_data = strong_labeler.label(sampled)
            if Mode == AL+ then
                train_set = concat(labelled, pseudo_labels)
            else then
                train_set = labelled
            end if
            model = train(model, train_set)
            evaluate(model, test_set)
    end for
end procedure
```

4.2 Proposed approaches in Detail

Mainly, there are three different active learning approaches that are compared against each other, all with minor differences. Additionally the performance of the 'Standard' mode that has been used to baseline of the model on the given data set.

1. Standard: Plain fully supervised learning on the chosen subset size with random sampling.
2. AL: The standard active Learning approach
3. AL+: Active Learning where the training set is the combination of annotated samples and pseudo labels. The pseudo labels are provided by the Weakly Labeler and are only accepted if the model predicted the same label with the given confidence threshold.
4. ALI: initial iteration is on the whole Weakly labels data set and the initially labelled instance and the further iterations are on the sampled samples

Overall, this framework introduces two novel methodologies, which are built upon the established concept of active learning in combination with a pre-existing large language model.
As active learning is a renowned approach it is further explained in the Background section 3.1. The aim of those two new approaches is to further improve the active learning in terms of minimizing the labelling effort.
The first, Active Learning Initial (ALI), distinguishes itself from standard active learning in that, during the initial step, the model is trained not only on randomly sampled instances but also on weakly labelled instances. However, in every subsequent step, conventional active learning is applied. This approach can be perceived as a form of warm-up training specific to the given data set. The second approach introduced is Active Learning Plus, which underwent several stages of development:
Initially, the entire data set was pre-labelled by a large language model, and then the model was trained on this complete data set. Consequently, the training set consisted mainly of weakly labelled instances and a minor portion of human-annotated instances. In each iteration, a small segment of these weakly labelled instances was sampled for training. As a result, the model's knowledge was largely influenced by the weak labeler, as only a minor percentage of the data set was properly labelled through the sampling method.
This led to the second stage of development, where a confidence threshold was introduced *(confidence = 1 - uncertainty)*. If the model achieved a certain confidence level on data instances, the corresponding weakly labels were accepted as additional pseudo labels in the training set. There is also an option to increase this confidence threshold consistently. If chosen to do so, the threshold is increased with each iteration if the number of selected pseudo labels exceeds that of the previous iteration, thus accommodating the model's increasing confidence. This concept was inspired by the paper [38].

However, this approach did not account for incorrect labels and accepted them blindly. Consequently, the final methodology was developed, which also takes into consideration the model's predictions. The model initially makes predictions on the remaining unlabeled training set, then it samples instances where it exhibits the greatest uncertainty. These instances will be strongly labelled subsequently. Next, instances are selected where the model exhibits sufficient confidence based on the threshold. Finally, for each of these instances, it is checked whether the weakly label matches the model's predicted class, i.e., the one with the highest probability. Only then is the instance accepted as a pseudo label and used in the combined training set. See 4.2 for a visualization.

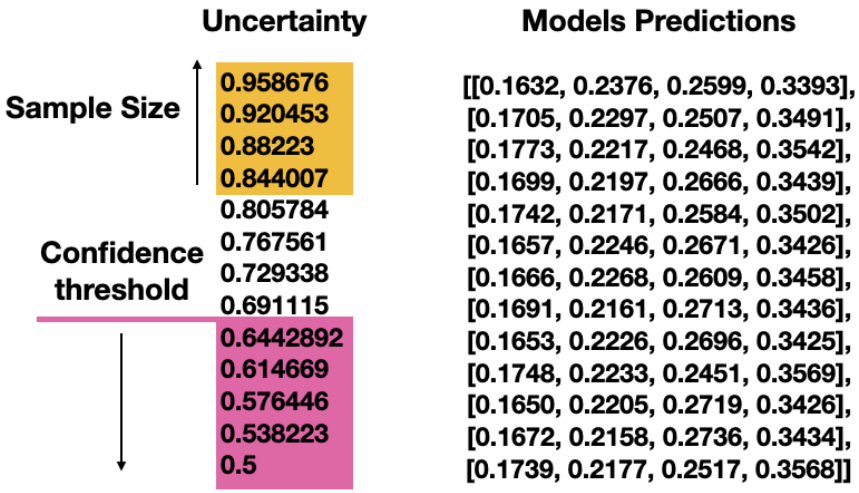


Figure 4.2: Sampling in Active Learning Plus

4.3 Data Set

The AG NEWS dataset [48] was chosen for this research. This is due to its widespread usage in various text classification studies. Moreover it is a widely used benchmark data set.

This a pre-labelled and balanced data set that is classified into 4 news categories: World, Business, Sport, Sci/Tech. It is provided in two parts training set that contains

120 000 instances and a test set that contains 7600 instances. It is divided into 3 columns: class index, Title, Description The data set is balanced over four classes.
For data preparation the pandas library was extensively used [24]. When preparing data the set it was split into 4 sub sets [table: very small, small, middle, big, very big, full]. So that a decision which set is to be used can be made at runtime.
The column Title is removed and the classes have been decremented by one. so the classes are encoded by the interval of [0,3] in the order they have been listed before. The Description column holds maximally 220 tokens (check).
In this study, two distinct approaches were considered for acquiring the initial sample. The first approach involved sampling the data randomly for each run, while the second approach assumed the presence of a smaller subset with annotations alongside a larger unlabeled portion of the data set.
The primary advantage of the first approach lies in its robustness, as it avoids reliance on a specific sample. This enhances confidence in the model's ability to generalize to unseen data.
On the other hand, the second approach capitalizes on the existence of a preexisting small annotated subset. However, one drawback is that the experiments do not exhibit substantial variation, with performance primarily assessed based on the model's learning ability.
For the experiments the choice has been made in favor of robustness thus the initial sample is random.

4.4 Model Training

The model employed in this study was 'bert-base-cased', a pretrained variant of BERT, accessed via the Huggingface Transformers library [42]. BERT is commonly utilized as a transfer learning model, and the user-friendly Huggingface library facilitates its straightforward implementation.
Being pretrained on a large corpus of text data, the model only requires fine-tuning on the final layer. It is also crucial to apply the BERT tokenizer to the input data to ensure compatibility with the model. In the context of this paper, the term "training" specifically refers to the fine-tuning process. The training parameters were selected based on the recommendations provided by the authors of BERT. According to the authors, it is recommended to utilize a batch size of 64 for the BERT base model and a learning rate of 5×10^{-5} during the fine-tuning process. Additionally, AdamW has been designated as the default optimizer for BERT [21]. Further reasoning Adam has been found to be performing well and is robust in machine learning applications [17]. Therefore, AdamW was chosen as the optimizer in the experiments conducted

in this study. For further reference on BERT, please refer the GitHub repository: `https://github.com/google-research/bert` [6].
To ensure fairness across all tested approaches, the early stopping technique was employed [29] [49]. The measure chosen for early stopping was the average validation loss of the model. The model was trained on the given data set until it ceased to show improvement. Since the training process is stochastic, it is possible for the loss to worsen from one training epoch to another. To prevent this, a patience counter of 3 was utilized. If the average validation loss did not improve over 3 consecutive epochs, the model with the best metric was returned via Model Checkpointing, and the next active learning iteration commenced. On the other hand, if the average validation loss improved, the counter was reset, and training continued. The maximum number of epochs was set at 500 to impose an upper limit.

4.4.1 Model Evaluation

The evaluation was carried out using two types of data sets: a validation set, where evaluation results contributed to the early stopping mechanism, and a test set, evaluated at each active learning iteration. Additionally, the best performance on the validation set was logged at each iteration. The techniques employed for evaluation include:

$$Accuracy = \frac{Number\,of\,Correct\,Predictions}{Total\,Number\,of\,Instances} \tag{4.1}$$

$$Recall = \frac{True\,Positives}{True\,Positives\,+\,False\,Negatives} \tag{4.2}$$

$$Precision = \frac{True\,Positives}{True\,Positives\,+\,False\,Positives} \tag{4.3}$$

$$F1\,Score = \frac{2\,\cdot\,Precision\,\cdot\,Recall}{Precision\,+\,Recall} \tag{4.4}$$

$$Average\,Validation\,Loss = \frac{\sum Validation\,Loss}{Total\,Number\,of\,Validation\,Instances} \tag{4.5}$$

The metrics used for evaluation in this study have been sourced from the 'Metrics for multi-class classification: an overview' paper by Grandini et al. [9] and another from another peer reviewed paper by Hossin et al. [13]. These are the most commonly used metrics for multi-class classification due to their comprehensive assessment of the models performance. Furthermore, these metrics have been recorded for each Active Learning iteration on the Validation and Test set. Additionally the average loss was also recorded on the average training set for training epoch. Moreover the best best performance on the validation set per AL iteration has also been logged. These metrics have been tracked along following measures: Epochs, AL iterations, Strong Labels.

To note the number epochs can greatly vary due to the employed Early stopping approach. For the AL+ mode the size (length) of the pseudo Labels has been tracked in addition to the error rate on the pseudo labels.

4.5 Experimental Set Up

To account for the computational budget, smaller experiments were employed to determine the optimal parameters that would be utilized for the large data set. The total labelling amount has been chosen to be 160 instances. The experimental runs have been conducted on 40 initial random samples and the rest picked by various technique , 12 instances per iteration over the period of 10 active learning (AL) iterations.
The focus in this work has been laid on uncertainty sampling in active learning. From this field 4 most commonly applied methods have been chosen [27]. In the following graph they are compared against each other with the help of the standard Active Learning approach without any modifications:

(a) Accuracy

(b) F1

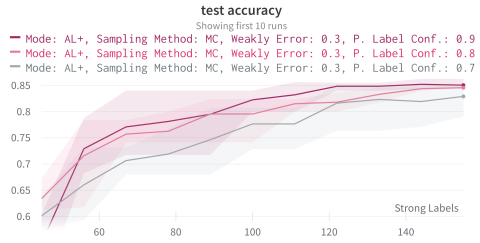
(c) Precision

(d) Recall

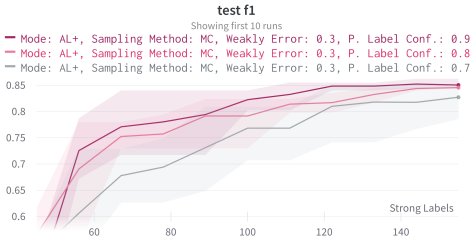
Figure 4.3: Evaluation of the uncertainty sampling techniques on a small set

The best sampling technique on 160 labels with active learning turned out to be Margin Of Confidence (MOC). It averages the maximum 82.5% test accuracy with the

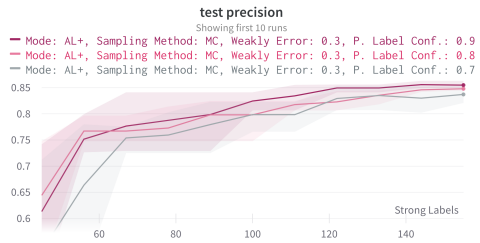
variance of 0.08 4.3(a). It has been primarily utilized as it is the most robust for multi class labeling in comparison to the other three implemented techniques. Even though you can see a similar performance across all the techniques, you can also see that MOC is the most robust this is indicative by the least variation in the experiments 4.3.
The decision to select a confidence threshold, similar to the sampling method, was based on experiments conducted on a smaller data set. In general, it's advisable to set a more stringent threshold for the AL+ approach to minimize potential errors in pseudo labelling. However, it's equally important to bear in mind that if the threshold is set too strictly, the process could revert to the conventional Active Learning method and performing extra computations that wouldn't have been necessary if Active Learning had been used from the beginning. As a reference authors of the paper "Cost-Effective Active Learning for Deep Image Classification" choose the confidence to be either 1 - 0.05 = 0.95 or 1 - 0.005 = 0.995 [38].
It's also worth noting the confidence threshold is examined by the combined performance over with the labelling error 30% ,which is the mean of the tested WERs. This provides insights into the robustness of the particular threshold. The resulting graphs from these small experiments are as follows 4.4.



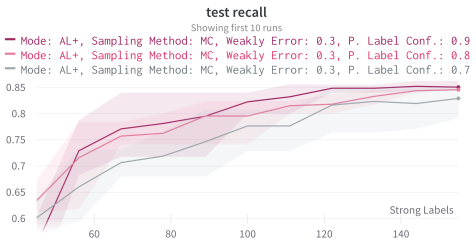
(a) Accuracy



(b) F1



(c) Precision



(d) Recall

Figure 4.4: Evaluation of different confidence thresholds on a small data set with Weakly Error Rate of 30%

The confidence threshold of 0.9 clearly outperforms the others in every evaluation metric 4.4(a), 4.4(b), 4.4(c), 4.4(d). Therefore the confidence threshold 0.9 has been chosen as the parameter for the large-scale experiments. The large-scale experiments were performed using the Holdout technique with a training set consisting of 40,000 instances and a full test set comprising 7,600 instances. The frequency of metric logging was reduced to once per Active Learning (AL) iteration. Similar to the smaller experiments, only 5% (1,600 instances) of the training subset were sampled, excluding the validation set. The validation set contained a total of 8,000 instances.

For the experiment, a patience counter of three was used, indicating that if the performance did not improve for three consecutive AL iterations, the process would terminate. The confidence threshold was set to 0.9, meaning that only instances with a confidence score of 0.9 or higher were considered for sampling.
Initially, the first AL iteration involved randomly sampling 1% (320 instances) from the total data set. Subsequently, in each AL iteration, 0.5% (160 instances) of the total training set were sampled. The experiment was conducted over 8 AL iterations in total.

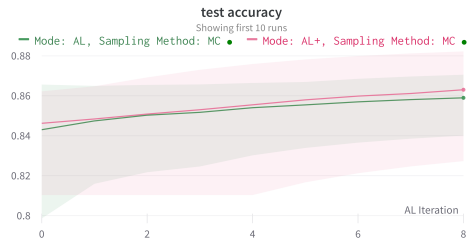
5 Results

5.1 Graphs

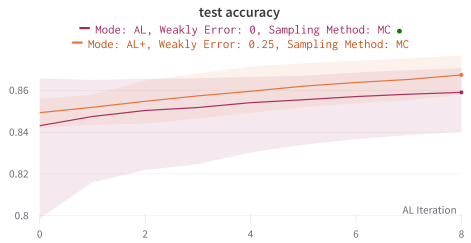
The various graphs presented employ color coding and sometimes dotted lines for simpler differentiation among them. The variance (depicted by min/max boundaries) across different experiment runs is depicted by a more transparent coloring around the line graphs. The runs are aggregated on their collective mean.
For consistency, the X-Axis depicts either strongly labelled data or Active Learning Iteration. Given that the sampling size remains constant in each AL iteration (except the initial sampling), this choice does not significantly affect the interpretation. While epochs could have been considered, the number of epochs can vary substantially due to the early stopping approach, making the chosen perspective more comprehensive and easily interpreted. The majority of the graphics and logging of the results have been accomplished using the WandB library [2].

5.2 Observations

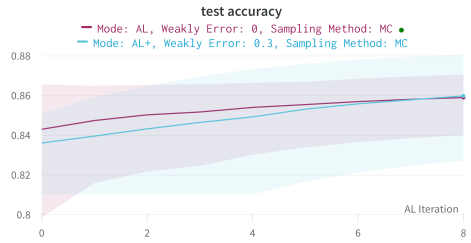
5.2.1 Active Learning vs Active Learning Plus



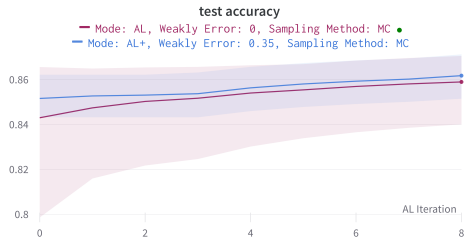
(a) All combined



(b) Weakly Error Rate: 25%



(c) Weakly Error Rate: 30%



(d) Weakly Error Rate: 35%

Figure 5.1: Comparison of Active Learning and Active Learning Plus

The figure depicts a comparison between Active Learning (AL) and Active Learning Plus (AL+) techniques. The first subfigure (Figure 5.1(a)) presents the overall performance of AL and AL+ combined, irrespective of the Weakly Error Rate (WER). The subsequent subfigures illustrate the performance of AL and AL+ at specific WERs: 25% (Figure 5.1(b)), 30% (Figure 5.1(c)), and 35% (Figure 5.1(d)).

5.2.2 Active Learning Plus Pseudo Labels

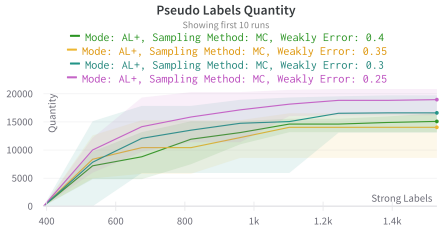


Figure 5.2: Quantity of Pseudo Labels

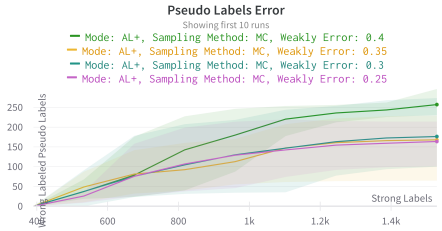


Figure 5.3: Erroneous Pseudo Labels

The first graph (Figure 5.2) illustrates the total number of labels the Active Learning Plus (AL+) approach draws from the training set. Among these, there are instances of erroneous pseudo labels, the quantity of which is depicted in the right graph (Figure 5.3).

5.2.3 Active Learning vs Active Learning Initial

Active Learning Initial Outlier

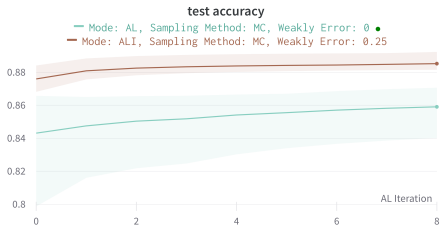


Figure 5.4: Runs without the outlier

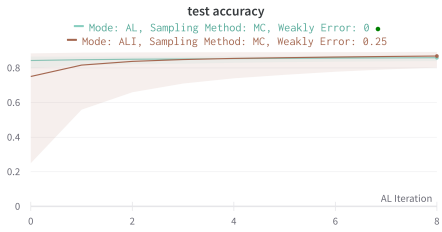
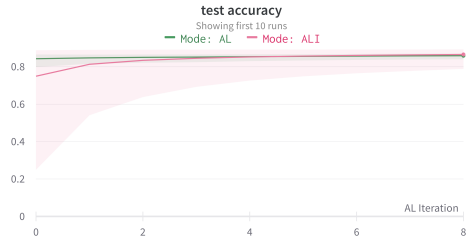


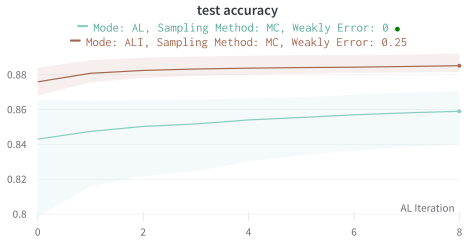
Figure 5.5: All five runs

The illustrated graphs depict the runs of Active Learning Initial (ALI) technique. The first graph (Figure 5.4) shows the combined performance of four runs with the Weakly Error Rate (WER) parameter set to 25%. The second graph (Figure 5.5) includes all five runs, but it contains an outlier that significantly affects the evaluated performance.

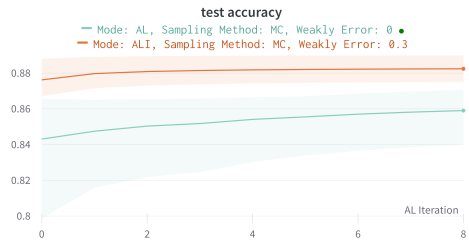
Consequently, for the subsequent graphs, the outlier has been excluded, and only the results from the four runs without the outlier are presented.



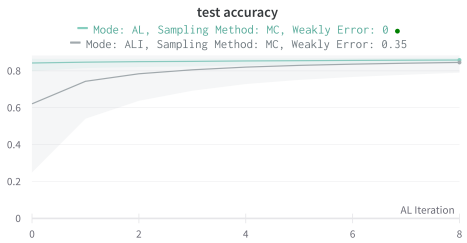
(a) All combined



(b) Weakly Error Rate: 25%



(c) Weakly Error Rate: 30%

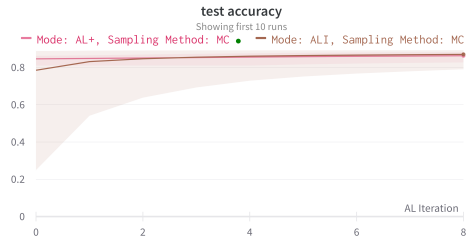


(d) Weakly Error Rate: 35%

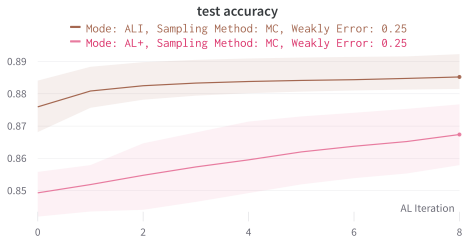
Figure 5.6: Comparison of Active Learning and Active Learning Initial

The four graphs compare the performance of standard Active Learning (AL) and Active Learning Initial (ALI). The second graph (Figure 5.6(b)) was previously mentioned. The first graph (Figure 5.6(a)) illustrates the comparison of the combined Weakly Error Rate (WER) between ALI and AL. It is evident that ALI outperforms AL consistently, but its performance is more dependent on the specific WER values.

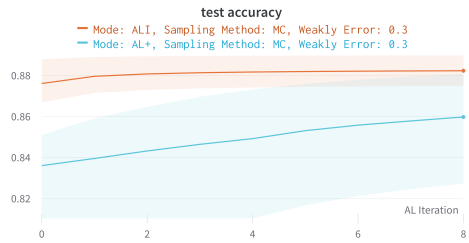
5.2.4 Active Learning Plus vs Active Learning Initial



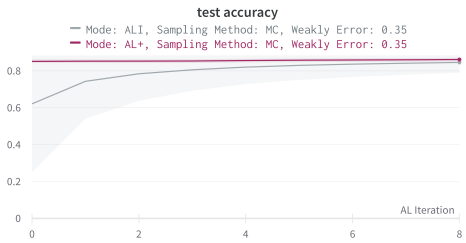
(a) All combined



(b) Weakly Error Rate: 25%



(c) Weakly Error Rate: 30%



(d) Weakly Error Rate: 35%

Figure 5.7: Comparison of Active Learning Plus and Active Learning Initial

The figure presents a comparison between Active Learning Plus (AL+) and Active Learning Initial (ALI) approaches. The first subfigure (Figure 5.7(a)) shows the combined performance of both approaches. It can be observed that ALI slightly outperforms AL+. However, when examining specific Weakly Error Rates (WER), the superiority of ALI becomes more evident. Figures 5.7(b) and 5.7(c) depict the performance at WERs of 25% and 30% respectively, where ALI demonstrates a clear advantage over AL+. These findings highlight the improved performance of ALI compared to AL+ in various scenarios.

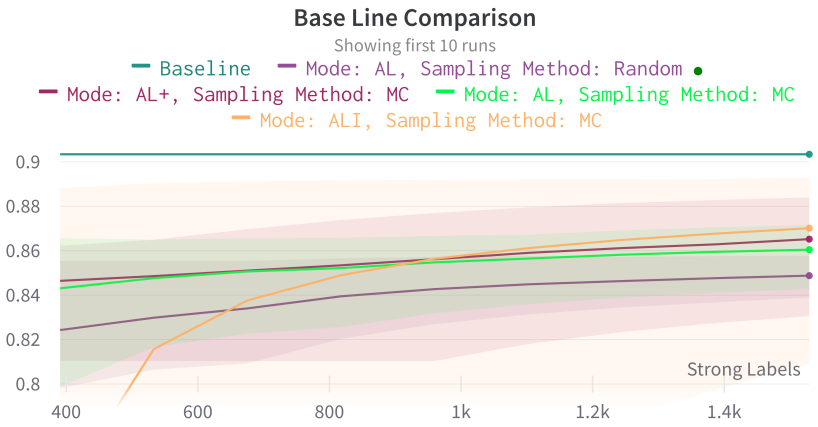


Figure 5.8: Base Line Comparison

5.2.5 Base Line Comparison

As a baseline, the Active Learning approach with random sampling was chosen, along with the standard training approach using the entire training set of 32000 instances. This was done to determine the maximum achievable test accuracy. The average accuracy on the entire data set was found to be 90.304%. This serves as a reference point for evaluating the performance of the proposed methodologies and comparing their effectiveness in minimizing labeling costs and improving accuracy.

5.2.6 Computational Cost

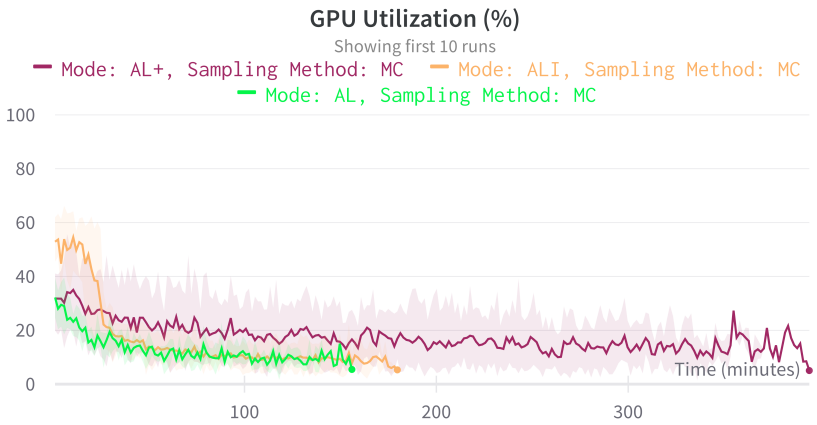


Figure 5.9: GPU utilization

This graphic portrays the percentual GPU utilization over time across AL, AL+ and ALI.

6 Discussion

Across all Weakly Error Rates (WERs) (5.1(b), 5.1(c),5.1(d)), AL+ consistently outperforms traditional AL. The figure provides a visual representation of the superior performance of AL+ over AL in various scenarios.
Furthermore, the graphs suggest that the Weakly Error Rate is not the sole determinant of the performance for this approach, as evidenced by the fact that AL+ with WER: 35% outperforms AL+ with AL+ with WER: 30%. However, one must consider the influence of the initially sampled set on the results of active learning, especially given it was randomly sampled.
Given this example, by observing the initial performance, it's apparent that the performance has higher variability for AL+ 30%, along with a generally lower starting point. However, it was not possible to identify an outlier in the AL+ 30% runs, as in the case of the ALI approach. Therefore, all five runs have been kept aggregated.
Considering the results, it is reasonable to hypothesize that the threshold at which the model accepts pseudo labels plays a critical role in this approach. It functions as a safeguard, only considering labels that match the weakly label predicted by the model. Yet, looking at the Pseudo labels graphic, it's apparent that reducing the weakly label error rate increases the total number of data instances sampled, while decreasing the number of wrongly labelled data. Despite the higher frequency of label errors with a higher WER, these seem rather insignificant. For example, even at the highest rate, AL+ 40 had at most 296 wrongly labelled instances. Even when considering 13179, the run with the fewest pseudo labels, the maximum error would be 0.226%. Therefore, it may be reasonable to consider using AL+ even with a weaker Large Language model.
However, it's important to note that the quantity of pseudo labels doesn't necessarily significantly improve the accuracy of the model. This might be due to these sampled pseudo labels not being informative enough, as they are the exact opposite of what uncertainty sampling techniques would have sampled, as seen in the image from the Methodology section 4.2.
The graphs 5.6 comparing Active Learning Initial (ALI) and AL reveal a significant difference, although it diminishes as the Weakly Error Rate (WER) increases. This suggests that the performance of ALI is highly dependent on the WER. In general, these approaches are quite similar. Their distinction lies in the initial data set: AL uses random sampling, while ALI employs random sampling and the remaining training

set is labelled by the large language model. Subsequently, the model is trained, and from that point on, both approaches align. This initial training helps the model obtain a broader overview, and afterwards, the model is only trained on selected samples. This could essentially be viewed as a form of mini pre-training, followed by fine-tuning on accurate and informative instances. One of the significant advantages of this approach is its simplicity and lower overhead compared to the AL+ approach.
This is further demonstrated in the computational cost diagram (5.9). It is evident that ALI heavily utilizes the GPU initially, but this usage swiftly declines to match the GPU usage of AL. In contrast, AL+ consistently exhibits higher GPU usage and takes significantly more time to perform its tasks.
The baseline is defined by standard non-iterative training, where early stopping is also implemented, similar to other approaches. As previously pointed out, Active Learning, employing the Margin of Confidence uncertainty sampling method, was clearly surpassed by both the Active Learning Plus and Active Learning Initial methodologies. Figure 5.8 offers a visual demonstration of these findings.
Additionally, Active Learning's random sampling approach is plotted for comparison. Interestingly, this approach yields surprisingly strong results, which may be attributed to the simplicity of the AG NEWS data set. AG NEWS is viewed as a relatively straightforward data set due to its balanced nature and clear-cut classes, presenting less ambiguity compared to more complex data sets, such as those used in sentiment analysis.

7 Conclusion

7.1 Limitations and Constraints

In the field of machine learning, particularly in the context of active learning, research is often guided by empirical exploration and iterative refinement, which poses challenges in providing universally applicable guidelines that are clear-cut and definitive. Each encountered problem exhibits unique characteristics and intricacies that require a customized approach [22]. Moreover, the complexity is amplified by the multitude of hyperparameters involved.
As a result, the solutions presented in this study may not be readily generalizable to other problems. It is important to note that the findings and conclusions are based on testing conducted specifically on a subset of the AG NEWS data set. The effectiveness and applicability of the proposed approaches may vary when applied to different data sets or problem domains.
This acknowledgment of the limitations and contextual dependencies is crucial in understanding the scope and validity of the presented results. Future research and experimentation are necessary to further validate these findings. With this in mind, it is important to acknowledge that this research has been constrained by time limitations and the availability of computational resources. As a result, only five runs were feasible for each specific configuration. Tracking the runs over the entire length of the data set was not possible. Instead, the analysis was limited to the 5% of labelled instances from the total training set. Despite these resource constraints, the research was able to make relative comparisons.

7.2 Conclusion

Regarding the central research question: "Is it possible to minimize the annotation workload by combining a weak annotator with active learning, rather than exclusively utilizing active learning?" the answer is affirmative. Both proposed methods surpass the performance of Active Learning utilizing the same sampling technique. While ALI outperforms Active Learning by a relatively larger margin, AL+ demonstrates greater resilience with regard to the Weakly Error Rate.

To answer the second main question: "What is the threshold for harnessing a Large Language model to leverage data from it, considering its weakly error rate?". A more extensive examination is necessary to give a confident answer, especially for the Active Learning Plus approach, given its coupled to two parameters: the threshold and the error rate in weak labeling.
As for ALI, accurately estimating the Weakly error until which ALI is profitable is challenging due to the limited number of runs below 35%. Therefore, it is difficult to provide an exact threshold at which employing ALI becomes worthwhile. However, based on quantitative analysis, a suggested recommendation is to consider ALI if the accuracy of the larger language model on the data set surpasses 70%. This benchmark ensures a significant probability of performance improvement.

7.3 Future Work

Although this research offers valuable insights, there are still some unexplored aspect which require future examination. First, instead of using plain random sampling, another subtype of sampling called cluster sampling can be beneficial for the initial sample. This approach, as proposed by Kang et al. [16], utilizes cluster-based sampling to select the initial training set for active learning in text classification. Similarly to random sampling, this approach does not require prior knowledge of the data set, although it entail additional overhead and knowing the number of classes in the data set can be beneficial.
Alternatively, in addition to the uncertainty methods, one should consider diversity sampling methods. Two diversity sampling methods were implemented, excluding random sampling, but they have not been thoroughly tested yet. Since AG NEWS is one of the simpler textual data sets, utilizing these methods may yield more satisfactory outcomes. In addition, one should consider combining these methods as proposed by R. Munro in his book [27].
Thirdly, exploring the two newly proposed methods would be interesting in a more challenging domain. For instance, uncertainty sampling performs well in such scenarios, as demonstrated in the paper "Cost-Effective Active Learning for Deep Image Classification" [38], where challenging image data is used. Therefore, one should consider choosing a more challenging textual data set than AG NEWS, for further exploration.
Moreover, in future work, it is planned to implement weight-based training, wherein instances labeled weakly are assigned less weight compared to those annotated by humans, referred to as strong labels.

List of Figures

List of Tables

Bibliography

[1] William H Beluch et al. "The power of ensembles for active learning in image classification." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9368–9377.

[2] Lukas Biewald. "Experiment tracking with weights and biases, 2020." In: *Software available from wandb.com* 2.5 (2020).

[3] Tom Brown et al. "Language models are few-shot learners." In: *Advances in neural information processing systems* 33 (2020). https://openai.com/blog/chatgpt/, pp. 1877–1901.

[4] Aron Culotta and Andrew McCallum. "Reducing labeling effort for structured prediction tasks." In: *AAAI*. Vol. 5. 2005, pp. 746–751.

[5] Ido Dagan and Sean P Engelson. "Committee-based sampling for training probabilistic classifiers." In: *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 150–157.

[6] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[7] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. "On Statistical Bias In Active Learning: How and When To Fix It." In: *arXiv e-prints* (2021). Published at ICLR 2021, arXiv–2101.

[8] Santiago González-Carvajal and Eduardo C Garrido-Merchán. "Comparing BERT against traditional machine learning text classification." In: *arXiv preprint arXiv:2005.13012, Journal of Computational and Cognitive Engineering* (2020).

[9] Margherita Grandini, Enrico Bagli, and Giorgio Visani. "Metrics for multi-class classification: an overview." In: *arXiv preprint arXiv:2008.05756* (2020).

[10] Steven C. H. Hoi et al. "Batch Mode Active Learning and Its Application to Medical Image Classification." In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 417–424. ISBN: 1595933832. DOI: 10.1145/1143844.1143897. URL: https://doi.org/10.1145/1143844.1143897.

[11] Steven CH Hoi, Rong Jin, and Michael R Lyu. "Batch mode active learning with applications to text categorization and image retrieval." In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1233–1248.

[12] Steven CH Hoi et al. "Batch mode active learning and its application to medical image classification." In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 417–424.

[13] Mohammad Hossin and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." In: *International journal of data mining & knowledge management process* 5.2 (2015), p. 1.

[14] Ming-Hui Huang and Roland T Rust. "Artificial intelligence in service." In: *Journal of service research* 21.2 (2018), pp. 155–172.

[15] Mohammad Hossein Jarrahi, Ali Memariani, and Shion Guha. "The Principles of Data-Centric AI (DCAI)." In: *arXiv preprint arXiv:2211.14611* (2022).

[16] Jaeho Kang, Kwang Ryel Ryu, and Hyuk-Chul Kwon. "Using cluster-based sampling to select initial training set for active learning in text classification." In: *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings 8*. Springer. 2004, pp. 384–388.

[17] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015* (2014).

[18] Elmar Kotter and Erik Ranschaert. "Challenges and solutions for introducing artificial intelligence (AI) in daily clinical workflow." In: *European Radiology* 31.1 (2021), pp. 5–7.

[19] Dong-Hyun Lee et al. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. 2013, p. 896.

[20] David D. Lewis and William A. Gale. "A Sequential Algorithm for Training Text Classifiers." In: *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (1994). Corrected version: 'A sequential algorithm for training text classifiers: Corrigendum and additional data', pp. 3–12. URL: `https://arxiv.org/pdf/cmp-lg/9407020.pdf`.

[21] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization." In: *arXiv preprint arXiv:1711.05101* (2017).

[22] David Lowell, Zachary C Lipton, and Byron C Wallace. "Practical obstacles to deploying active learning." In: *arXiv preprint arXiv:1807.04801, Presented at Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2018).

[23] Katerina Margatina et al. "Active learning by acquiring contrastive examples." In: *arXiv preprint arXiv:2109.03764* (2021).

[24] Wes McKinney et al. "Data structures for statistical computing in python." In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[25] Lester James Miranda. "Towards data-centric machine learning: a short review." In: *ljvmiranda921. github. io* (2021).

[26] Thahir P Mohamed, Jaime G Carbonell, and Madhavi K Ganapathiraju. "Active learning for human protein-protein interaction prediction." In: *BMC bioinformatics* 11.1 (2010), pp. 1–9.

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

[28] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. "Pervasive label errors in test sets destabilize machine learning benchmarks." In: *arXiv preprint arXiv:2103.14749, Presented at NeurIPS 2021* (2021).

[29] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[30] Mamshad Nayeem Rizve et al. "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning." In: *arXiv preprint arXiv:2101.06329, Presented at ICLR 2021 Conference* (2021).

[31] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. "Active hidden markov models for information extraction." In: *Advances in Intelligent Data Analysis: 4th International Conference, IDA 2001 Cascais, Portugal, September 13–15, 2001 Proceedings 4*. Springer. 2001, pp. 309–318.

[32] Andrew I Schein and Lyle H Ungar. "Active learning for logistic regression: an evaluation." In: (2007).

[33] Burr Settles. "Active learning literature survey." In: (2009).

[34] Burr Settles. "From Theories to Queries: Active Learning in Practice." In: *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*. Ed. by Isabelle Guyon et al. Vol. 16. Proceedings of Machine Learning Research. Sardinia, Italy: PMLR, 16 May 2011, pp. 1–18. URL: https://proceedings.mlr.press/v16/settles11a.html.

[35] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. "Query by committee." In: *Annual Conference Computational Learning Theory*. 1992.

[36] Masashi Sugiyama and Shinichi Nakajima. "Pool-based active learning in approximate linear regression." In: *Machine Learning* 75 (2009), pp. 249–274.

[37] Ashish Vaswani et al. "Attention is all you need." In: *Advances in neural information processing systems* 30 (2017).

[38] Keze Wang et al. "Cost-Effective Active Learning for Deep Image Classification." In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2017), pp. 2591–2600. DOI: 10.1109/TCSVT.2016.2589879.

[39] Liantao Wang et al. "Active learning via query synthesis and nearest neighbour search." In: *Neurocomputing* 147 (2015), pp. 426–434.

[40] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[41] H James Wilson and Paul R Daugherty. "Collaborative intelligence: Humans and AI are joining forces." In: *Harvard Business Review* 96.4 (2018), pp. 114–123.

[42] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[43] Hao Wu and Saurabh Prasad. "Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification." In: *IEEE Transactions on Image Processing* 27.3 (2018), pp. 1259–1270. DOI: 10.1109/TIP.2017.2772836.

[44] Xing Wu et al. "COVID-AL: The diagnosis of COVID-19 with deep active learning." In: *Medical Image Analysis* 68 (2021), p. 101913.

[45] Bishan Yang et al. "Effective multi-label active learning for text classification." In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 917–926.

[46] Daochen Zha et al. "Data-centric artificial intelligence: A survey." In: *arXiv preprint arXiv:2303.10158* (2023).

[47] Cha Zhang and Tsuhan Chen. "An active learning framework for content-based information retrieval." In: *IEEE transactions on multimedia* 4.2 (2002), pp. 260–268.

[48] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In: *Advances in neural information processing systems* 28 (2015).

[49] Wangchunshu Zhou et al. "Bert loses patience: Fast and robust inference with early exit." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18330–18341.