Active Learning for Multi-label Text Classification with Transformers

Master Thesis

presented by
Emanuela Kuhlman
Matriculation Number 1580703

submitted to the
Data and Web Science Group
Dr. Ines Rehbein
University of Mannheim

March 2022

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Acknowledgements

Thank you to my supervisor, Dr. Ines Rebhein, for her guide, patience and support over the duration of my research.
Thank you to my family and particularly my husband for his critical feedback.

Chapter 2

Introduction

Motivation

Today, the amount of data created or captured to fuel machine learning is growing rapidly in every industry. (add citation) This increase of data creates a hidden cost with current methods due to expensive and time-consuming manual data labeling which is tedious and prohibitive.

The main types of ML methods deployed are supervised learning, unsupervised learning, and reinforcement learning. Of the methods, supervised learning has been widely adopted by many enterprises because of its ability to do more with less. For some industries, this method is not completely feasible because the complexity in the labeling process increases in domains of specified expertise, which require experts or educated human judgment. The result is increased cost, time, and effort that ultimately leads to a bottleneck in the operation. A technique called Active Learning offers a direct solution to this bottleneck by selecting only the most informative examples for labeling and training of the model. In this paper, I analyze the efficiency and practicalities of using Active Learning in addressing Multi-Class labeling constraints in text classification problems.

Active Learning (AL) (also called "query learning" or sometimes "optimal experimental design" in the statistics literature) is a sub-field of machine learning which queries data instances to be labeled for training by an oracle (e.g. a human annotator).[Settles, 2009]

AL can be particularly useful in enabling the adoption of Machine Learning to leverage the big pools of unlabeled data. The theory behind this success is that an active learner can freely choose which instance to query and the model learns from the feedback obtained from the oracle, which results in superior learning performance with less labeled data.[Settles, 2009]

AL has been applied to many real-world problems in natural language processing (NLP) [Thompson et al., 1999], [Tang et al., 2002], [Tür et al., 2003], image retrieval [Cheng and Wang, 2007], [Liu et al., 2008], [Zhang et al., 2008], image classification [Qi et al., 2008],[Joshi et al., 2009], named entity recognition [Hoi et al., 2006], remote sensing [Li and Guo, 2013] [Tuia et al., 2012], text categorization [Wang et al., 2017] [Mccallum and Nigam, 1998] [Zhu et al., 2010], recommender systems [Saito et al., 2015] [Atighehchian et al., 2020] [Karimi et al., 2011], visual question answering [Lin and Parikh, 2017] and object detection [Qu et al., 2020] [Aghdam et al., 2019].

Text classification is the shining example amongst NLP problems where AL application has yielded outstanding performance. A domain expert manually labelling training data requires significantly more effort for multi-label text data since the expert needs to decide all possible labels for one data point. Recent uses of Transformer Architectures in NLP displayed state-of-the-art results advancing the field, particularly in the area of text classification. (cite Transformers: State-of-the-Art Natural Language Processing) A wide array of research literature exists in AL, focusing on both single and multi-label text classification tasks. Despite this breadth research, there are no widely available research that examines AL with transformer models for the task of multi-label text classification. My thesis will address this specific method and is structured as follows: Firstly, I lay out the problems this thesis addresses 2. Next, I provide the definitions to the central concepts of the thesis 3 and review related work 4. Then I describe my approach, present the experiments, and analyze the results 5. Finally, I discuss the limitations and possible future work.

Problem Statement

My research in this thesis tests the question of feasibility of combining AL with transformer models for the task of multi-label text classification. In other words, it addresses if it is beneficial to use AL when a labeling budget is very small while retaining the powerful capabilities of BERT-like models. I look to multi-label classification instead of single because the benefits would be significant given the cost of an expert needed to spend the time on deciding all the possible labels that could apply to one data point. This is particularly an issue when labeling expansive text covering several pages and documents.

My focus here is practical real-world scenarios of multi-label text classification. Choosing the right classifier/learner in Machine Learning is challenging. This is even more difficult in the AL setting because:

1. The right learner needs to be selected by using a very small amount of labeled data.

2. The learners role is twofold: to make predictions and to work with a query strategy which helps refine it.

3. Choosing the right multi-label text classifier faces an output space being the power set of the set of classes, as it exponentially grows with the number of classes.

4. The growth of the output space more than often translates to a high class imbalance with the distribution of data per each class is unequal. Furthermore, characteristics of multi-label data sets like cardinality and density have been proven to be related to the difficulty of learning a multi-label classifier. [Bernardini et al., 2014]

Another facet of this research is a systematic investigation of different sampling strategies for pool-based active learning with a focus on uncertainty-based methods. Uncertainty based query strategies are a computationally inexpensive alternative. Despite their relative disadvantages in traditional active learning, when paired with transformers, they are highly effective as well as efficient. [Schröder et al., 2021] In classical AL strategies the prediction probability of a model is used as a representation for how confident the models is. Deep neural networks are very complex and have many parameters. It is not a good predictor of model uncertainty because it tends to produce probabilities with a very high confidence.
With this in mind, I compared different uncertainty selection strategies to assess their capabilities of choosing samples that are good representatives of the distribution of the pool of unlabeled data.

Chapter 3

Theoretical Framework

Preliminaries

Introduction to Active Learning

AL is an iterative process that takes advantage of the collaboration between humans and machines to select a small subset of data to label. The developer builds an initial baseline classification machine-learning model to predict the outputs for all the unlabeled data. The model iteratively queries the human annotator to label data that according to a query strategy deemed most informative. Then, the developer adds newly labeled data to the previously labeled data and uses it to improve the model. This labeling and learning process repeats until the limited annotation budget is exhausted.

There are several scenarios to pose active learning queries. Figure 1 illustrates the three main types that include:

- **Membership Query Synthesis**, where the learner generates its new artificial instances from an underlying natural distribution. [Angluin, 1988] This scenario is reasonable for many finite problem domains (Angluin, 2001). Membership query synthesis for natural language processing tasks may create streams of text that produces gibberish and labeling such arbitrary instances can be awkward when the oracle is a human annotator. [Settles, 2009]

- **Stream-Based Selective Sampling**, a learner receives distribution information from the environment and queries an oracle on parts of the domain it considers “useful”[Cohn et al., 1994], i.e the unlabeled instances are queried one at a time and active learning algorithms have to decide whether or not to ask a human expert to label it.

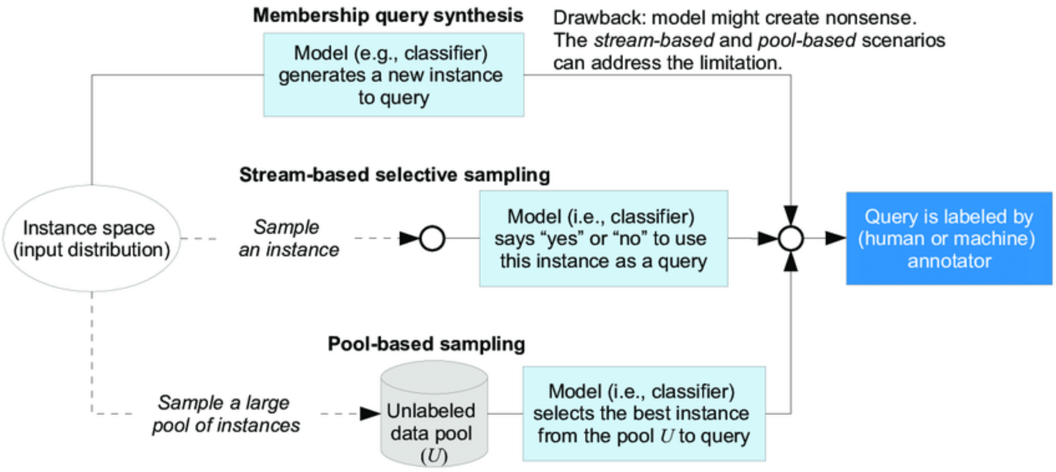

Membership query synthesis
Model (e.g., classifier) generates a new instance to query
Drawback: model might create nonsense. The *stream-based* and *pool-based* scenarios can address the limitation.
Stream-based selective sampling
Instance space (input distribution)
Sample an instance
Model (i.e., classifier) says "yes" or "no" to use this instance as a query
Query is labeled by (human or machine) annotator
Pool-based sampling
Sample a large pool of instances
Unlabeled data pool (U)
Model (i.e., classifier) selects the best instance from the pool U to query

Figure 3.1: An illustration of the three main active learning scenarios [Yang et al., 2018]

- **Pool-Based Sampling** is where the instances are selected from the entire data pool and usually assumed to be closed (i.e., static or non-changing), based on an informativeness measure used to evaluate all the instances in the pool. [Lewis and Gale, 1994] Stream-based active learning algorithms can perform worse than pool-based methods[Ganti and Gray, 2012]. There are more frequent data point queries for human annotation in the stream-based setting than that in the pool-based methods. One reason is that in the stream-based setting, AL algorithms cannot go through all the unlabelled data to select the most useful samples. It is likely that the annotation budget is already finished before the most informative samples appear in the stream. Most active learning algorithms focus on the pool-based scenario.

The advantage of stream-based strategies lies on its computational efficiency because there is no need to go through the data pool to query the best sample. This approach reduces annotation effort and limits the size of the database used in nearest-neighbour learning which expedites the classification algorithm making it more efficient. [Settles, 2009] The price of high efficiency is a weaker performance.

The different methods of active learning start by measuring the informativeness of unlabelled instances and select(sample) new instances that support the learning process. There are various selective sampling frameworks for active learning.

Amongst the most popular ones are:

- **Query-by-committee(QBC)** selects a committee of classifiers by randomly sampling hypotheses from the version space. [Seung et al., 1992] QBC constructs a committee consisting of a number of different classifiers/models (always more than two). The next unlabeled example is selected by the principle of maximal disagreement among these classifiers, i.e., each committee member can vote on the labeling of unlabeled instances and the instance which causes greatest disagreement within the committee will be selected for annotation.

- **Expected Model Change** - identifies the instance that would impart the greatest change to the current model if we knew its label.[Settles et al., 2008a] The intuition behind this framework is that it prefers instances that are likely to most influence the model, i.e., have greatest impact on its parameters regardless of the resulting query label. This approach has been shown to work well in empirical studies, but can be computationally expensive if both the feature space and set of labels are very large. [Settles, 2009] The differences among these approaches lies in the criteria to measure the model change. Settles et al. [Settles, 2009] proposed to measure the expected gradient length of the objective function. Freytag et al. [Freytag et al., 2014] estimated the change of model outputs instead of model parameters. Cai et al. [Cai et al., 2017] proposed to use the gradient of the loss function to approximate the model change and derived algorithms for both SVM and logistic regression classifier.

- **Uncertainty Sampling** in this framework, an active learner queries the instances about which it is least certain how to label.[Lewis and Gale, 1994] Uncertainty sampling searches for unlabeled data points that are near the decision boundary and it is least certain how to label. The selected unlabeled examples are used to clarify the position of decision boundaries.

Uncertainty Sampling

Among the most popular AL strategies, that can also be used in a Deep Learning setting is Uncertainty Sampling. Extensive research has been conducted on how to estimate and quantify uncertainty.

There are few uncertainty measure tools including:

- **Entropy** A more general uncertainty sampling strategy uses entropy [Shannon, 1948] as an uncertainty measure. Entropy is an information-theoretic measure that

represents the amount of information needed to "encode" a distribution. As such, it is often thought of as a measure of uncertainty or impurity in machine learning.
Approaches based on entropy, select samples with maximum entropy, defined as:

$$x^* = argmax_{x \in U} - \sum_{y \in C} P_L\left(y_i|x;\theta\right) logP_L\left(y_i|x;\theta\right) \tag{3.1}$$

where $P_L\left(y_i|x;\theta\right)$ is the conditional probability of y given x according a a classifier trained on L. For binary classification, entropy-based uncertainty sampling is identical to choosing the instance with posterior closest to 0.5. However, the entropy-based approach can be generalized easily to probabilistic multi-label classifiers and probabilistic models for more complex structured instances, such as sequences [Settles et al., 2008b]

- **Least Confidence** queries instances for which the model is least certain according to the max-entropy decision rule [Lewis and Gale, 1994]

$$x^*_{LC} = argmin_{x \in U} P_L\left(y^*_i|x;\theta\right) \tag{3.2}$$

- **The Smallest Margin** is a popular active learning sampling method by Scheffer et al. [Scheffer et al., 2001] that selects candidates with the smallest margin, where the margin is defined as follows.

$$x^* = P_L\left(y^*_1|x;\theta\right) - P_L\left(y^*_2|x;\theta\right) \tag{3.3}$$

Multi-label text classification

Definition 1 *Assuming a given nonempty input space X of documents and a predefined finite set of class labels $\{1, \ldots, n\}$. The target space Y in multi-label classification is defined as $Y \ =^{def} \ \phi(\{1, ..., d\})$, where $\phi(A)$ represents the power set of a given set A. The task of multi-label classification is that of estimating a target function $f : X- > Y$, based on a given training set of labeled examples [Angluin, 1988]:*

$$L = \{(x_1, Y_1), \ldots (x_m, Y_m)\} \subset X * Y$$

Various approaches have been proposed in the literature for solving multi-label problems: [de Carvalho and Freitas, 2009]

- Combining single-label classifiers to deal with the multi-label classification task

- Modifying single-label classifiers by adapting their internal mechanisms to allow their use in multi-label

- Designing new algorithms to deal with multi-label problems.

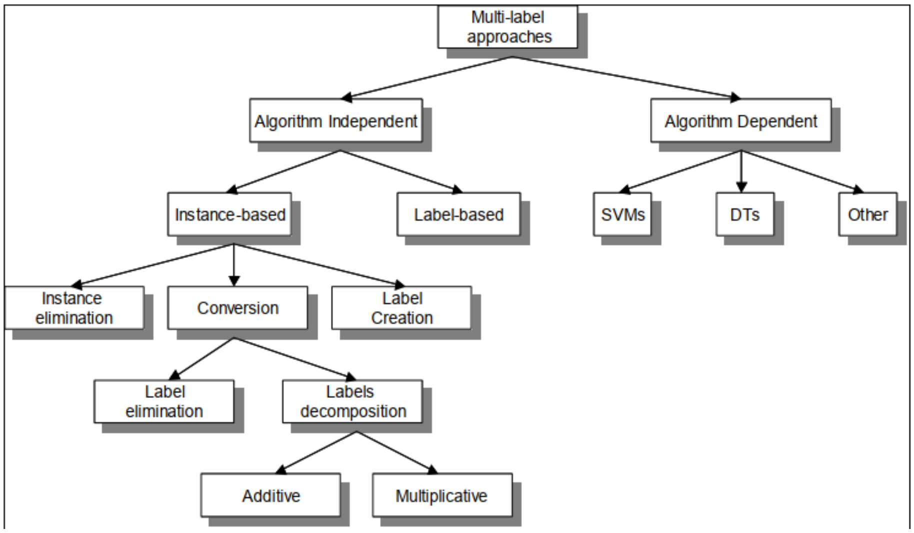


Figure 3.2: Methods used in Multi-Label Classification Problems [de Carvalho and Freitas, 2009]

The different methods, displayed in Figure 3, can be grouped in two main approaches: Algorithm Dependent and Algorithm Independent.

One popular algorithm independent approach for multi-label text classification used for research from [Esuli and Sebastiani, 2009], [Yang et al., 2009], [Brinker, 2006] is a method which transforms the data to fit to multi-class algorithms. The approach is of transforming the original problem into a series of single-label problems, also known as Binary Relevance (BR) or one-vs-the-all/ one-vs-the-rest. The generalized binary relevance decomposition was first employed by Joachims (1998) [Joachims, 1998] with support vector machines for the binary subproblems.

One binary decomposition method used to solve multi-label problems is training a set of separate binary classifiers $h_i : X- > \{-1, +1\}$ for each of the n target labels against the rest set of labels. [Boutell et al., 2004]. Training the binary classifier h_i leads to relabelling of all examples $(x, Y) \in L$ as positive if $i \in Y$, and the rest as negative. Target objects Y for unseen patterns x are predicted according to positive classification of the underlying set of binary classifiers (ALL-positive):

$$h : X \to Y \text{[Boutell et al., 2004]}$$

$x \to argpos_{1,...,n}h_i(x) = \{i \in \{1, ..., d\} | h_i(x) = +1\}$ [Brinker, 2004]

The algorithm dependent methods focus on modifying algorithms to perform multi-label classification.

Transformers

Natural language processing models have achieved state-of-the-art results using transformer-based deep pre-trained models like BERT (Bidirectional Encoder Representations from Transformers). The performance of these Language Models (LM) has been enhanced by their use of multi-headed attention mechanisms and the fact that they are pretrained on a large-scale corpus. [Vaswani et al., 2017]

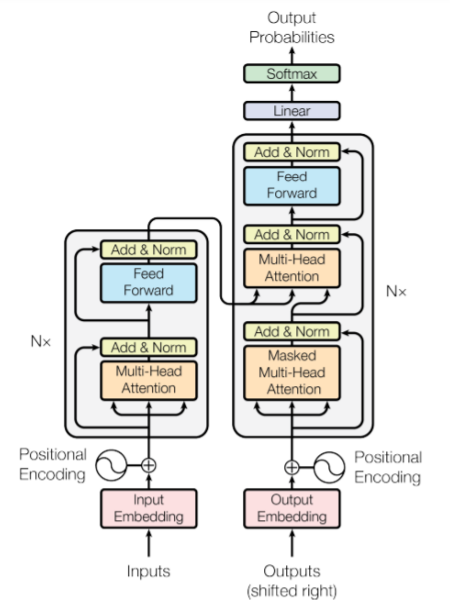


Figure 3.3: Transformers Architecture
[Vaswani et al., 2017]

BERT-like models produce representations of each token using only the transformer encoder part (left part on Figure architecture). The encoder stacks multiple transformers layers which are composed of multi-head attention mechanisms and Feed-Forward sub-networks. Each transformer layer incorporates self-attention across the inputs and residual layers/normalization/fully connected layers within each input. The multi-headed attention mechanism is used to learn contextual representation by attending multiple times on same inputs instead of learning left-to-right and right-to-left sequence representation like Recurrent Neural Networks (RNN). The learned representation then flows into a classification layer. To adapt transformer models in the multi-label classification setting, the sigmoid activation function can be used instead of softmax. This way for each of the labels the independent probabilities is predicted.

Masked Language Models (MLM) like BERT have hundreds of millions of parameters can be very complex and training them on small and imbalanced data sets can be complicated. This challenge has been tackled by first pre-training BERT on large corpora on the domain of interest like BookCorpus(800M words) [**?**, **?**], English Wikipedia (2,500M words), clinical notes and PubMed Central articles, and then the model is fine-tuned on the smaller data set [Devlin et al., 2018]. Other state-of-the-art MLM share the idea of BERT but vary in design. ELECTRA-Efficiently Learning an Encoder that Classifies Token Replacements Accurately [3] on the other side is a discriminator model. ELECTRA Large like BERT consists of 24 layers, 1024 hidden units and 336M parameters. Clark et al. [Clark et al., 2020] found that a discriminator enables training even with limited computing resources. DistilRoberta is a smaller, light-weight model. It consists of six layers, hidden units of size 768, and 82M parameters and offers competitive performance with a lower computational cost [Sanh et al., 2019a]

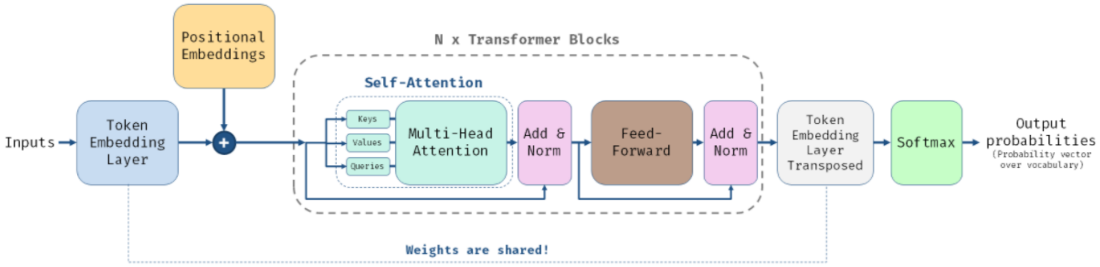


Figure 3.4: BERT Architecture
[Ponzetto, 2021]

Chapter 4

Literature Review

The first empirical study on the combination of AL strategies with BERT was conducted by Ein-Dor et al. [Ein-Dor et al., 2020] The focus of their work were various binary classification tasks. To understand the impact of AL in the performance when using BERT-based models they explored a diverse set of AL strategies.
Random as the baseline, where data is randomly sampled from the large pool of unlabeled data.
Uncertainty-based sampling methods such as Least Confidence, Monte Carlo Dropout [Gal et al., 2017], and uncertainty sampling using ensemble methods - Perceptron Ensemple, Expected Model Change(Expected Gradient Length (EGL) - choosing the examples with the largest expected gradient norm, with the expectation over the posterior distribution of labels for the example according to the trained model. [Huang et al., 2016])
Diversity Sampling [Gissin and Shalev-Shwartz, 2019] (Discriminative Active Learning (DAL) selects instances that make the labeled set of instances indistinguishable from the unlabeled pool.
Core-Set - chooses the examples that best cover the data set in the learned representation space using farthest first traversal algorithm. [Sener and Savarese, 2018]

They performed a comparative analysis between the different strategies by measuring two batch properties that are known to impact the AL effectiveness, diversity and representativeness. AL strategies showed to further boost the performance of BERT, however an inconsistency in the performance of AL was observed, leaving an open question: Which selection strategies are more suitable for the combination of AL with models like BERT?

Transformer-based language models like BERT have been pre-trained on large text corpora, and they can be fine-tuned to a specific tasks using less training data than when trained from scratch. These models have proven to be highly performant but they have a high number of model parameters which makes them computationally very expensive for query strategies that are targeted at text classification [Settles, 2009].

Research on different query strategies for AL is quite rich, among which are uncertainty-based approaches. Sampling through uncertainty is quantified through predictive uncertainty which in deep neural networks consists of two parts: data uncertainty (aleatoric) [Der Kiureghian and Ditlevsen, 2009] and model uncertainty(epistemic). [Kendall and Gal, 2017]

Nguyen et al. [Nguyen et al., 2022] analyzed different uncertainty measures and quantification with a focus on different epistemic and aleatoric uncertainty methods. They showed that epistemic uncertainty sampling in the sense of uncertainty sampling based on measures of epistemic uncertainty in a prediction shows strong performance and consistently improves on standard uncertainty sampling. [Nguyen et al., 2022] Defining the right acquisition function, i.e., the condition on which a sample is most informative for the model, is the main challenge of AL-based methods.

Gal et al. [Gal et al., 2017] exercises Dropout to sample weights to approximate the posterior distribution over labels, and use it to detect samples that reduce model uncertainty. Ein-Dor et al. [Ein-Dor et al., 2020] also infered from their studies that the two uncertainty quantification strategies are computationally more expensive than others: Dropout, because of the larger number of inference cycles, and the Expected Gradient Length strategy since the gradients depend on the model and when used as a query strategy, they scale with the vast number of a transformer's parameters, and they need to be computed per-instance and not batch-wise. [Schröder et al., 2021]

[Schröder et al., 2021] analyzed two uncertainty-based query strategies combined with transformers, Prediction Entropy and Breaking Ties. Through simulated experiments they showed that these strategies slightly improved the accuracies over the random baselines. [Ash et al., 2020] adopted a combination of uncertainty and diversity approach and introduced Batch Active learning by Diverse Gradient Embeddings (BADGE). BADGE computes gradient embeddings g_x for every candidate data point x in U and then uses clustering to select a batch. Each g_x is computed as the gradient of the cross-entropy loss with respect to the parameters of the model's last layer. [Ash et al., 2020]

Yuan et al. (2020) [Yuan et al., 2020] focused on a cold-start of AL with BERT. They developed a strategy called Active Learning by Processing Surprise (ALPS) and BERT K-Means which combine uncertainty and diversity sampling. ALPS acquisition uses the masked language model (MLM) loss of BERT as a representation of model uncertainty in the downstream classification task. ALPS forms a surprisal embedding s_x for each x,by passing the unmasked input x through the

BERT MLM head to compute the cross-entropy loss for a random 15% subsample of tokens against the target labels. ALPS clusters these embeddings to sample k sentences for each AL iteration. BERT K-Means on the other hand, works similarly to ALPS with the difference that it uses BERT embeddings.[Yuan et al., 2020]

Zhang et al. [Zhang et al., 2020] designed a new query strategy Margin, Intra-correlation and Inter-correlation (MII) based on posterior probabilities which combines uncertainty with instance correlation to actively select instances. The performance of MII was checked in combination with BERT and proved to be an effective approach, however no information regarding the execution time.

Chapter 5

Experimental Evaluation

Data

I experimented with the various AL query strategies and evaluated their performance on five large representative data sets used for text classification.
The selected corpora are used for legal document classification and web-based communication sentiment classification.

The first data set I used is the Multi-Eurlex text data set, which has been used as a benchmark data set to evaluate multi-label text classification algorithms. Multi-Eurlex is a multilingual data set introduced by [Chalkidis et al., 2021] It comprises 65k European Union laws translated into 23 EU official languages. available at https://huggingface.co. Each law of the data set has been labeled with one or more EUROVOC concepts. [Chalkidis et al., 2021] provided three label sets comprising 21, 127, 567 EUROVOC concepts respectively. Due to computational limitations, I experimented with level one containing 21 classes. Use of the Multi-Eurlex data sets was motivated by the size and availability in 23 different languages. It allowed for the evaluation of AL on a large data set and for comparison of results across languages such as English, German and Spanish.

The second data set is GoEmotions, a large human-annotated data set of 58k Reddit comments extracted from popular English-language subreddits and labeled with 28 emotion categories (including neutral). [Demszky et al., 2020] I used the version filtered based on reter-agreement, which contains a split into train, test and validation with 43k, 5k and 5k samples correspondingly.

For comparison reasons, I extended my experiments from initially using the whole MultiEurlex corpus to also including a self generated synthetic data set, MiniEurlex, that contains the 10 most frequent categories in the data set. MiniEurlex is smaller with a total of 54k instances and has 10 distinct labels.

Some important characteristics of multi-label data sets related to the distribution of labels are:

-**Label cardinality** :the average number of labels per instance

$$Card = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{5.1}$$

-**Density** :the mean of the number of labels of the instances that belong to the data set divided by the number of data set's labels

$$Dens = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|} \tag{5.2}$$

-**Label set** :the number of distinct label sets

An overview of the data sets used for validation of AL strategies with the total number of instances, languages, label cardinality and density is shown in table 4.1. The differences between the selected data sets rely not only characteristics of multi-label data set like label mentioned above, but also on the input length. The number of tokens on the MultiEurlex documents varies from 200 to over 3000 posing a challenge for model like BERT that have a max length limit of 512 tokens. The maximum sequence length in training and evaluation datasets of Go_Emotions is 30.

data set	Language	Instances	Nr. Distinct Labels	Labelset
Emotions	English	58000	28	5
MultiEurlex	English	65000	21	9
MultiEurlex	German	65000	21	9
Multi_Eurlex	Spanish	65000	21	9
Mini_Eurlex	English	54207	10	7

Table 5.1: An overview of the data sets

Multi-label data sets are mostly very imbalanced, meaning that some labels are very frequent and others are more rare. There are also instances that might have been assigned a combination of a each of these labels with different distribution in a data set.

To measure the level of imbalance of the data sets, I used the tools introduced by [Charte et al., 2015]:

- The mean imbalance ratio (MeanIR) calculated as:

$$MeanIR = \frac{1}{M}\sum_{i=1}^{M} IRLbl(y_i) \tag{5.3}$$

- and the coefficient of variation of CVIR :

$$CVIR = \frac{\sqrt{\frac{\sum_{i=1}^{M}(IRLbl(y_i)-MeanIR)^2}{M-1}}}{MeanIR} \tag{5.4}$$

where IRLbl(Li) is the level of imbalance of a label, calculated as the ratio of the most frequency of the most frequent label with the frequency of the current label.

double check again CVIR values!!

Table 5.2: Imbalance Metrics for each data set

Data set	Cardinality	Density	MeanIR	CVIR
GoEmotions	1,177	0,042	36,5	1,61
MultiEurlex	3,232	0,154	453,8	2,54
MiniEurlex	2,723	0,272	58,3	0,99

IRlb is equal to 1 for the label that is more frequent in the data set and the rest of the labels have IRLb > 1. MultiEurlex data set has the highest imbalance level with the value of the MeanIR being 453 times higher than the ideal value of MeanIR (ideally is 1) and variance over 100% 254% (ideally 0%). The level of imbalance of GoEmotions and the synthetically generated also remains very far from the ideal balanced data set scenario.

The numbers obtained by the calculations (Table 4.2) show the difference in the imbalance levels between each of the data sets and will help with the interpretation of the performance of AL. Figure 4.1 and 4.2 shows a visualization of the distribution of the labels and labelsets for each data set.

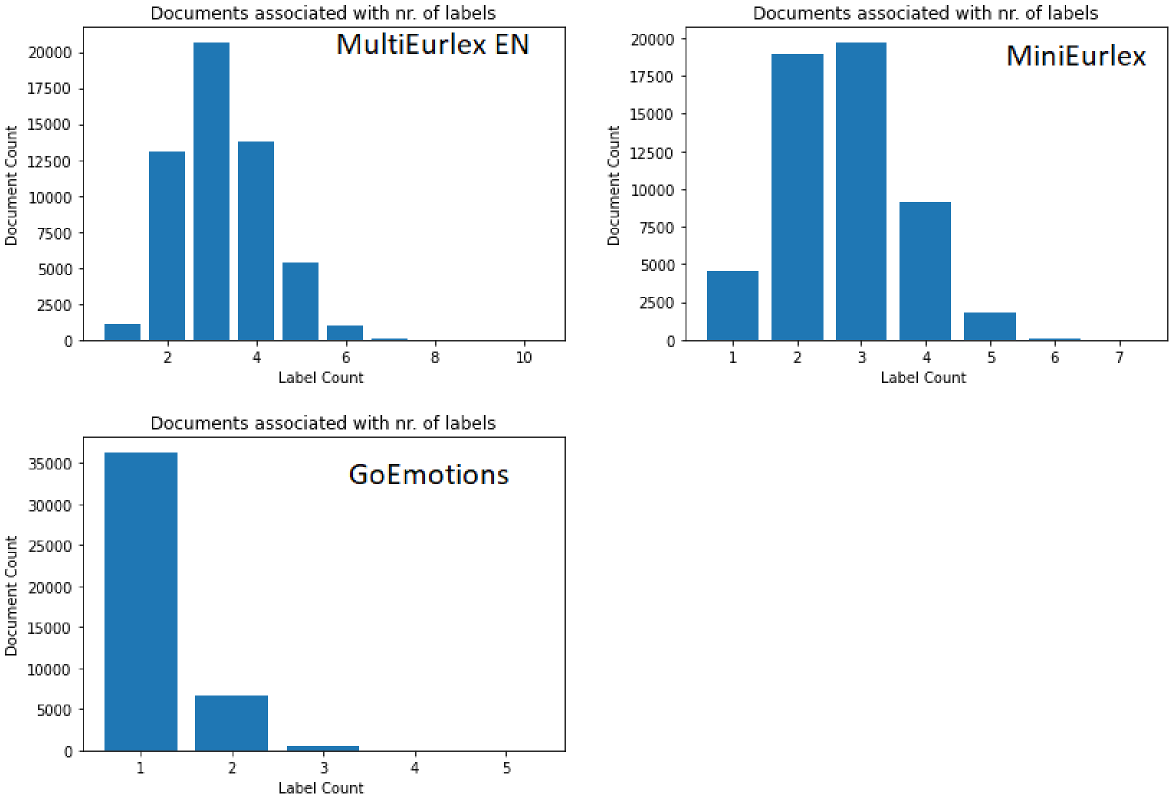


Figure 5.1: Labelset distribution for each data set.

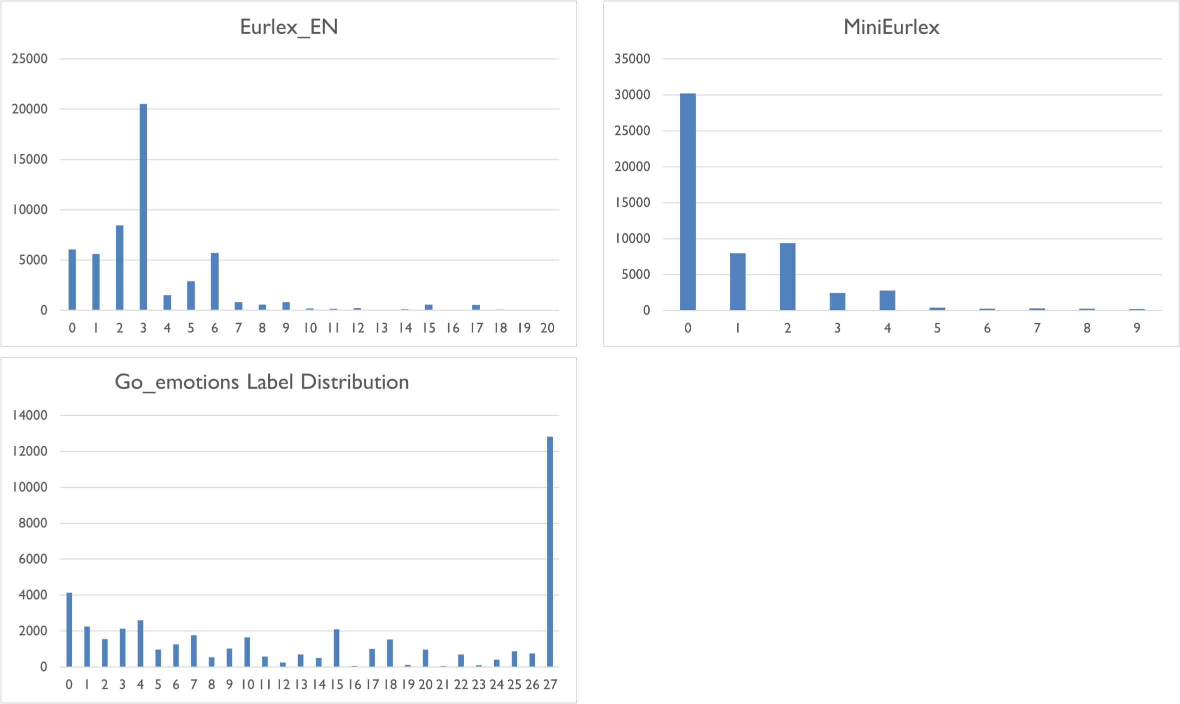


Figure 5.2: Label distribution for each data set.

5.1 Settings

Training & AL Strategies

I adopted the pool-based AL approach in batch mode. The assumption of pool-based AL is that there is a small pool of labeled data L and a large pool of unlabeled data U. [Settles, 2009]. The process begins by training a small amount of data from L to initialize a transformer based multi-label text classification model M. The model consists of a transformer model and a classification layer on top of it. The classification layer has n output neurons which correspond to each label. A query strategy uses M to choose some instances from U which are then labeled by a human annotator. The labeled instances are then added to L and the model is trained using all the labeled data available.

Inspired by the recent work of [Ein-Dor et al., 2020] on AL for BERT, I fine-tuned three transformer models:

- BERT [Devlin et al., 2019] considered the current baseline of NLP tasks.

Algorithm 1 Pool based Active Learning

```
Input: Pool of labeled data L
Pool of Unlabeled data U
Number of labels n
Classifier M
Number of iterations I
Sample selection size s
for $i = 1$ to $I$ do
    $M \leftarrow$ Train data in L
    $S \leftarrow$ Select s data points from U
    $s \leftarrow$ Human annotates s-data
    $< s, l > \leftarrow$ Add to L
end for
```

- DistilRoberta [Sanh et al., 2019b] which offers competitive performance with a lower computational cost.
- DistilBert[Sanh et al., 2021] a lighter distilled version of BERT.

As suggested by [Hu et al., 2018], to avoid overfitting of data, fine-tuning is done from scratch on each AL iteration.
To understand the impact on AL on the performance, I experimented with the following strategies: Least Confidence- (Lewis and Gale, 1994) selects instances with the least prediction confidence regarding the most likely class according to the max-entropy decision rule. Prediction Entropy-(Shannon, 1948) selects samples with the highest entropy in the predicted label distribution with the aim to reduce overall entropy. Breaking Ties-Selects instances which have a small margin between their most likely and second most likely prediction.[Luo et al., 2004] and Bert-KMeans- a density based query strategy.(add references!!) As in previous works by [Ein-Dor et al., 2020], I compared the performance of each strategy to a Random Sampling strategy. Experiments were performed on a single Nvidia's Tesla P100 with 16GB of RAM. The implementation was an adaptation of the code made available on the library Small-Text [Schröder et al., 2021].

Initial Training Set

For a warm start of AL and train the first model an initial training set needs to be selected. I experimented with different sizes 100, 500 and 800: I set the initial size on the Emotions data set 800 and on the MultiEurlex 500. I conducted the majority of the experiments following a similar experimental setup as [Yang et al., 2009] and with an initial training set of 500 examples for the Eurlex data set.

To build a representative first training set I used the algorithm proposed by

Sechidis, Tsoumakas and Vlahavas [Sechidis et al., 2011] called Iterative Stratification. This algorithm stratifies a multi-label data set by using the subsets sampling approach. The classes are considered individually and are distributed starting from the classes that are more rare and working up to the classes that are more frequent.

One other important parameter for AL is the number of new instances that will be selected on each iteration of AL from the unlabeled pool. I experimented with a sample size of 100 samples for 10 iterations and 50 samples for 20 iterations for one of the experiments (The corresponding results are on Appendix B). The selected samples are added to the labeled pool and transformers were retrained from scratch for 10 epochs. To obtain the results in Figure XXX I use as budget the 3.4% of the data pool for the MultiEurlex data set for each of the languages and 2.7% of the data pool for Go_emotions.

Annotation The instances selected by each query were labeled before they were added to the pool of labeled data for the next training of the classifier. In a real-world scenario, an expert of the field would be annotating these data. Under lack of available expertise on the domains of my experiments, I used the labels already available on the data sets before they were removed to form the Unlabeled data pool.

Results

In this section I initially present the measures I used to evaluate the performance of classification on MultiEurlex, GoEmotions and MiniEurlex, and a measure that facilitates the comparison of each implemented AL strategy.

Model Evaluation

For the evaluation measure of the performance of each classification model I used the standard Micro-Average F1 evaluation score used in text classification research since it tends to be more informative for multi-label classification tasks. The F1-score is the harmonic average of the precision and recall (R) defined as:

$$F_1 = \frac{2PR}{P+R}\text{[Yang, 2001]}$$

The F1-score is micro-averaged, i.e. all true positives, false positives and false negatives are collected in a joint pool and then the recall, precision and F1 values are computed from that pool.[Yang, 2001]

I computed the average of micro-F1 scores for every active learning iteration over three random experiments. Table 4.3 summarizes the classification results of circa 150 experiments measured by micro-F1 after 10 active learning iterations. Additional results are available on Appendix B.

Table 5.3: Micro-F1 score on the data sets with 1000 training samples added (%) using DistilBERT

Data set	RS	LC	PE	BT
GoEmotions	0,43	0,45	0,41	0,47
MultiEurlex EN	0,73	0,75	0,75	0,71
MultiEurlex DE	0,74	0,74	0,75	0,70
MultiEurlex ES	0,76	0,77	0,75	0,73
MiniEurlex EN	0,82	0,81	0,80	0,80

Query Strategy Evaluation

According to the definition of Reyes et al. [Reyes et al., 2018] of the ideal selection strategy, a selection strategy is considered ideal if it is able to select in every iteration a set of unlabelled instances implying the construction of a classifier that it is superior to all classifiers generated in previous iterations. One of the most common ways of comparing different active learning strategies is visual. From the results of my experiments it is difficult to conclude if there is a significant difference in the performance of the strategies as their accuracy curves are overlapping very often.

To compare the active learning methods I used a deficiency measure used on other studies [Zhu et al., 2008]. It can be defined as:

$$Def_n(AL, REF) = \frac{\sum_{i=1}^{n}(acc_n(R) - acc_i(AL))}{\sum_{i=1}^{n}(acc_n(REF) - acc_i(REF))} \tag{5.5}$$

with acc_i being the average accuracy at the ith iteration, n the number of annotated instances, R the Random baseline method and AL the query strategies that will be compared. A deficiency value < 1.0 is an indicator of the query strategy performing better than the Random method and a value > 1.0 indicates a poor performance.

Table 5.4: Average deficiency achieved by various active learning methods. The stopping point is 1000.

Dataset	Model	LC	PE	BT
MiniEurlex EN	DistilBert	0,27	0,49	1,01
	DistilRoberta	0,92	0,94	1,69
MultiEurlex DE	DistilBert	0,66	0,73	1,15
MultiEurlex ES	DistilBert	0,39	1,02	1,46
GoEmotions	DistilBert	0,34	0,38	-0,16
	DistilRoberta	0,54	0,68	1,01
MiniEurlex EN	DistilRoberta	2,41	1,38	2,31

Discussion

The results of the experiments help verify the applicability of AL for the problem of multi-label classification. The two determining factors are achieving a high classification performance with minimal amount of data within a reasonable amount of time.

The curves in Figure 4.3 show the change in accuracy values as batches of samples are labeled on each iteration of AL. To facilitate comparison with performance of transformer models when the entire corpora are used, the best values for GoEmotions and MultiEurlex are plotted as a horizontal line. A best classification score for MiniEurlex is absent since this is a data set I synthetically generated.

The best Micro-F1 scores achieved from [Demszky et al., 2020] by training a BERT based model using the full Go_Emotions data set was 0.46. From the results on table 4.2 and 4.3 can be seen that by using AL the same accuracy score was achieved after after only 6 iterations using 33 times less training data. The low accuracy in short text classification is usually affected by the lack of contextual information. This however, is beyond the scope of this research.

The same performance as the state-of-the-art results were achieved on the Multi-Eurlex Data set within 5 iterations of AL while labeling only 1.8% of the training data.

Pretrained transformer models are well known for their very high performance, and the results achieved on all data sets are a very good indicator that their capabilities are maintained when they are being actively trained with small chunks of data. Very noticeable is the behavior of AL strategy with each transformer model. It is visibly clear that DistilRoberta outperforms DistilBert on MultiEurlex and GoEmotions as higher accuracies are achieved in less steps. Although DistilRoberta appears superior to DistilBert, the curves of the query strategies are overlapping very often with the Random sampling method, expect for BT on the Eurlex data

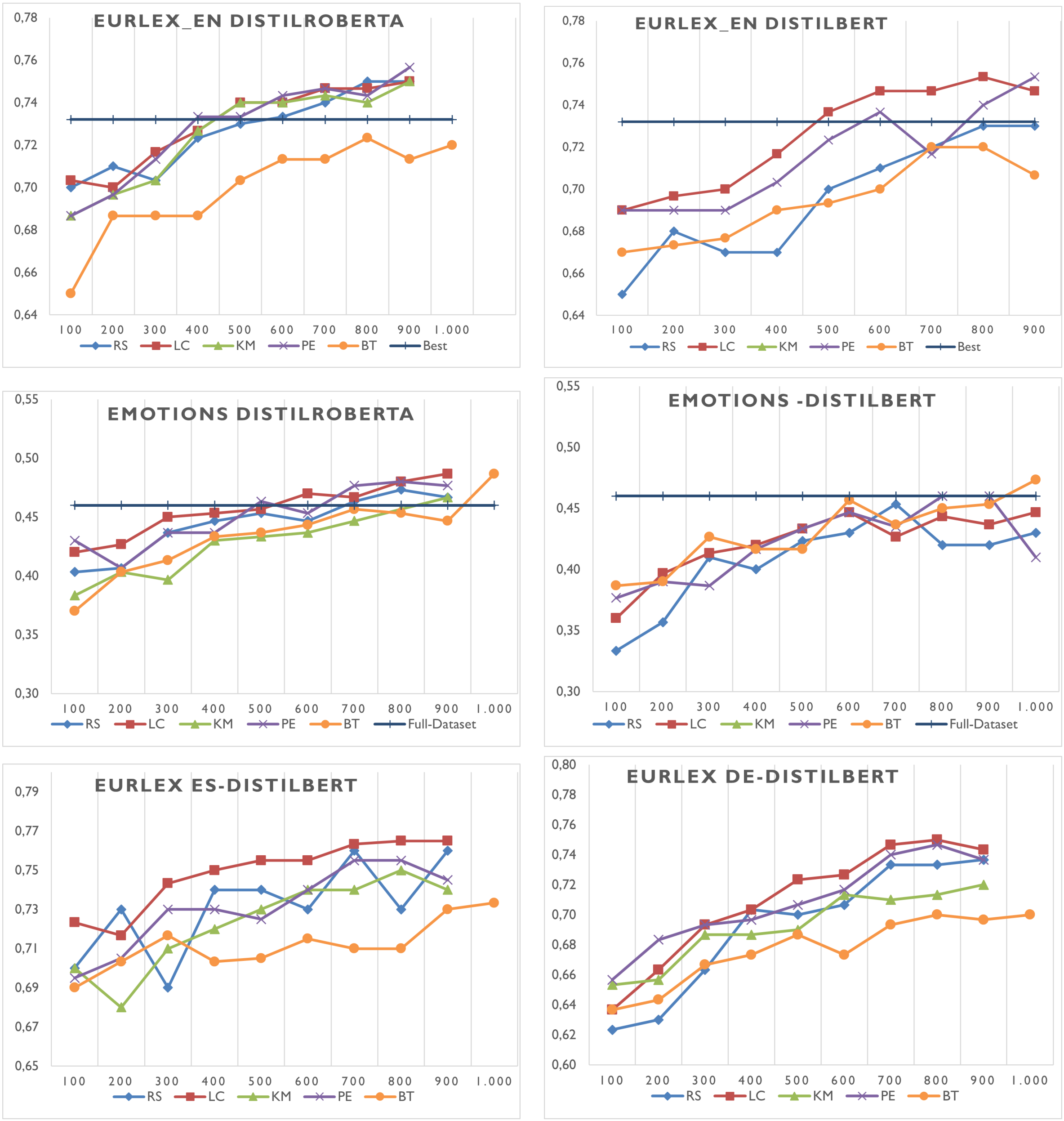


Figure 5.3: Test accuracy during AL iterations using Least Confidence, BERT K-Means, Prediction Entropy and Breaking Ties against Random. The plotted results are averaged across three runs.

set.

Runtime for one iteration of Al with BERT, when ran for 10 epochs on the MultiEurlex dataset was on average 10,3 min for RS and the uncertainty based methods and 18,3 minutes for K-Means. The difference on the runtime for one iteration of DistilBert and DistilRoberta were insignificant. The average 6,7 minutes for RS, LC, PE and BT and 9 minutes for K-means.

I experimented with different sample size (results are on Appendix B), and while a very large sample size of 1000 increases the runtime per iteration, the difference in runtime between selecting a sample of 50 or 100 instances was insignificant. This leads me to assume that the most computationally expensive step of the process is the inference, when the models is selecting the new data points for which to request labels for. For the query to decide on the next batch of samples from a large unlabeled data pool of approximatelx 55k data points on the MultiEurlex dataset requires more computations and time before moving on to the next iteration.

Choosing the best query strategy

To decide on the best performing strategy, I compared each of them with baseline random sampling through the help of the Deficiency Measure. The results of the comparison are displayed in Table 4.3. The deficiency values for each query strategy vary across data sets and transformer models. From the results, we can observe that the Least Confidence and Prediction Entropy achieved the lowest deficiency values. The deficiency measure of Least Confidence is 0,27 for the English dataset, 0,66 for German and 0,39 for Spanish. The values remain low on Go_emotions as well, 0,34 with DistilBert and 0,54 with DistilRoberta. The two strategies provide very promising performance on both data set in all three languages.

The highest deficiency values was the Breaking Ties(BT) strategy, which quantifies uncertainty by using the difference between the two highest posterior probabilities. Contrary from the results of [Schröder et al., 2021], BT is less effective on a multi-label setting with imbalanced data sets. After several iterations, it still reaches the same high accuracy scores as the other strategies and its performance remains very poor when compared to randomly selected samples. My results show that Least Confidence and Prediction Entropy are the most effective and robust methods, resulting in the selection of samples that are more representative of the entire data set.

K-Means is the most computationally expensive and does not provide any improvements over random sampling; this disadvantage became apparent on the very first iterations of AL. Due to its very poor performance, it is not further shown in other results. Unexpected outcome were the result found from training DistilRoberta on the same settings and dataset. While the deficiency measure of the strategies proportionally changes from one model to the other, the rate of change is very high which supports DistilBert's superiority.

AL with BR

Binary Decomposition Experiment One of my initial experimental approaches to handling multi-label classification was to treat it as a set of independent binary classification problems and train one binary model per label. The multi-label classifier is indicated in the following diagram.

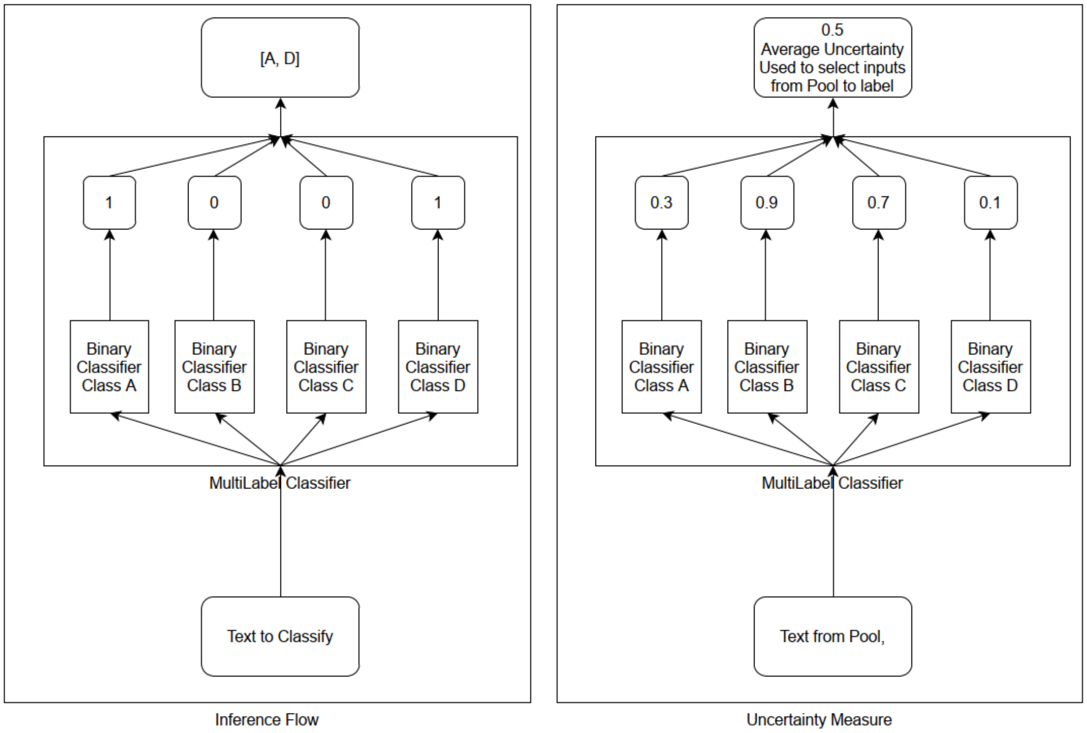

[A, D]
1
0
0
1
Binary Classifier Class A
Binary Classifier Class B
Binary Classifier Class C
Binary Classifier Class D
MultiLabel Classifier
Text to Classify
Inference Flow
0.5 Average Uncertainty Used to select inputs from Pool to label
0.3
0.9
0.7
0.1
Binary Classifier Class A
Binary Classifier Class B
Binary Classifier Class C
Binary Classifier Class D
MultiLabel Classifier
Text from Pool,
Uncertainty Measure

Figure 5.4: Binary Relevance

One Binary Classifier per target label would be trained. During AL, the uncertainty measures of each Binary classifier would be taken into consideration to decide which sample to label next. During Inference, each binary classifier is run on the input to determine whether the label for that classifier should be in the multilabel classifier output. This method is impractical for the combination of model and corpora I was using. Using the Binary Relevance approach made it very difficult to fine-tune a single instance of a transformer-based classifier with the recommended settings on a single GPU. Running it on the GPU with the EurLex was a computationally infeasible approach. Training 28 binary classifiers in parallel (for the Go_emotions data set), when each requires 2+ GPUs is prohibitively expensive. Using AWS SageMaker price estimates, a single binary classifier requires a $3hr instance. Training the full 28 category multilabel classifier would be roughly $270hour, making it unsuitable for experimentation or exploration.

Training from scratch instead of fine-tuning would allow the use of a much smaller network size, but would take an extreme amount of time to converge and would require many examples to get good performance (usually not a good fit for active learning), as well as being more limited in final performance than the pre-trained models.

Chapter 6

Conclusion

Supervised ML requires a significant amount of precisely labeled data. Manually labelling data sets, particularly assigning multiple labels, is a brute force approach that is not effective nor does it scale well when facing an increase in data size and number of labels. AL on the other hand reaches increased accuracy with less data and removes the need for this brute force approach. A crucial aspect for determining the suitability of using transformers with AL is not only achieving superior accuracy with less data, but also the time it takes to train them. Long run times defeat the purpose of practicality in use where humans are involved and waiting for the model to train.

My experimental results show that AL is an effective approach that can achieve superior accuracy with only 3Even though under both data sets AL performed equal or greater than models using the entire corpus, the run-time offset the savings in labeling cost. While AL does not need an entire corpora, the time of training the transformers increases with each iteration and often connected to the size of the unlabeled data pool.

The developer can manage or mitigate the cost benefit analysis of accuracy versus runtime when deciding the best query strategy independent of the data set and model. I found Density-based strategies such as K-mean demands longer runtimes in order to reach comparable accuracies. Uncertainty strategies perform best under time constraints, but is still impractical with human annotation. Of the uncertainty strategies, Least Confidence is the most effective and robust for training multi-label text classifiers than any of the other evaluated active learning methods.

The research in this thesis supports the conclusion that utilizing AL with transformers for multi-label text classification allows for enhanced model learning with limited amounts of labeled data while retaining the high performance capabilities offered by transformer based-models. The benefits achieved far outweigh any cost in time required for this method. The model's success with the difficult problems show promise for wider application.

Future Work

With this in mind, future work would be aimed towards reducing the runtime of AL. There are various directions that can be followed:

- Applying Parallel Active Learning a method introduced by [Haertel et al., 2010] to reduce the long waiting times with minimal staleness. The technique allows training and inference to happen in parallel while the expert is assigning labels to the selected instances.
- Given the link between the size of the unlabeled data pool and the amount of time the query strategy needs to select new instances from it, Subsampling is one effective technique that can be applied [Shi et al., 2020] to reduce the size of the unlabeled data pool.
- It might be interesting approaching multi-labels classification with a new strategy proposed to reduce the label space like Multi-label prediction via Compressed Sensing (CS) [Hsu et al., 2009] which assumes sparsity in the label set and encodes labels using a small number of linear random projectors.

Bibliography

[Aghdam et al., 2019] Aghdam, H. H., Gonzalez-Garcia, A., Lopez, A., and Weijer, J. (2019). Active learning for deep detection neural networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3671–3679.

[Angluin, 1988] Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2:319–342.

[Ash et al., 2020] Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. (2020). Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*.

[Atighehchian et al., 2020] Atighehchian, P., Branchaud-Charron, F., and Lacoste, A. (2020). Bayesian active learning for production, a systematic study and a reusable library. *CoRR*, abs/2006.09916.

[Bernardini et al., 2014] Bernardini, F., Silva, R., Rodovalho, R., and Mitacc Meza, E. (2014). Cardinality and density measures and their influence to multi-label learning methods. *Learning and Nonlinear Models*, 12:53–71.

[Boutell et al., 2004] Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771.

[Brinker, 2004] Brinker, K. (2004). Active learning of label ranking functions. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pages 129–136, New York, NY, USA. Association for Computing Machinery.

[Brinker, 2006] Brinker, K. (2006). On active learning in multi-label classification. In Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A., and Gaul, W., editors, *From Data and Information Analysis to Knowledge Engineering*, pages 206–213, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Cai et al., 2017] Cai, W., Zhang, Y., Zhang, Y., Zhou, S., Wang, W., Chen, Z., and Ding, C. (2017). Active learning for classification with maximum model change. *ACM Trans. Inf. Syst.*, 36(2).

[Chalkidis et al., 2021] Chalkidis, I., Fergadiotis, M., and Androutsopoulos, I. (2021). Multieurlex - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *CoRR*, abs/2109.00904.

[Charte et al., 2015] Charte, F., Rivera, A. J., del Jesus, M. J., and Herrera, F. (2015). Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163:3–16. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems Progress in Intelligent Systems Mining Humanistic Data.

[Cheng and Wang, 2007] Cheng, J. and Wang, K. (2007). Active learning for image retrieval with co-svm. *Pattern Recognition*, 40(1):330–334.

[Clark et al., 2020] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

[Cohn et al., 1994] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

[de Carvalho and Freitas, 2009] de Carvalho, A. and Freitas, A. (2009). *A Tutorial on Multi-label Classification Techniques*, volume 205, pages 177–195.

[Demszky et al., 2020] Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A. S., Nemade, G., and Ravi, S. (2020). Goemotions: A dataset of fine-grained emotions. *CoRR*, abs/2005.00547.

[Der Kiureghian and Ditlevsen, 2009] Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatoric or epistemic? does it matter? *Structural Safety*, 31(2):105–112.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Ein-Dor et al., 2020] Ein-Dor, L., Halfon, A., Gera, A., Shnarch, E., Dankin, L., Choshen, L., Danilevsky, M., Aharonov, R., Katz, Y., and Slonim, N. (2020). Active learning for BERT: An empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

[Esuli and Sebastiani, 2009] Esuli, A. and Sebastiani, F. (2009). Active learning strategies for multi-label text classification. In *In Proceedings of the 31st European Conference on Information Retrieval (ECIR 2009*, pages 102–113.

[Freytag et al., 2014] Freytag, A., Rodner, E., and Denzler, J. (2014). Selecting influential examples: Active learning with expected model output changes. In *ECCV*.

[Gal et al., 2017] Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep Bayesian active learning with image data. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

[Ganti and Gray, 2012] Ganti, R. and Gray, A. (2012). Upal: Unbiased pool based active learning. In Lawrence, N. D. and Girolami, M., editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 422–431, La Palma, Canary Islands. PMLR.

[Gissin and Shalev-Shwartz, 2019] Gissin, D. and Shalev-Shwartz, S. (2019). Discriminative active learning. *CoRR*, abs/1907.06347.

[Haertel et al., 2010] Haertel, R., Felt, P., Ringger, E., and Seppi, K. (2010). Parallel active learning: Eliminating wait time with minimal staleness. pages 33–41.

[Hoi et al., 2006] Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. (2006). Batch mode active learning and its application to medical image classification. ICML '06, page 417–424, New York, NY, USA. Association for Computing Machinery.

[Hsu et al., 2009] Hsu, D. J., Kakade, S. M., Langford, J., and Zhang, T. (2009). Multi-label prediction via compressed sensing. In *NIPS*.

[Hu et al., 2018] Hu, P., Lipton, Z. C., Anandkumar, A., and Ramanan, D. (2018). Active learning with partial feedback. *CoRR*, abs/1802.07427.

[Huang et al., 2016] Huang, J., Child, R., Rao, V., Liu, H., Satheesh, S., and Coates, A. (2016). Active learning for speech recognition: the power of gradients. *CoRR*, abs/1612.03226.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Joshi et al., 2009] Joshi, A. J., Porikli, F. M., and Papanikolopoulos, N. (2009). Multi-class active learning for image classification. In *CVPR*.

[Karimi et al., 2011] Karimi, R., Freudenthaler, C., Nanopoulos, R., and Schmidt-thieme, L. (2011). Non-myopic active learning for recommender systems based on matrix factorization. In *In IEEE Information Reuse and Integration (IRI). IEEE.*

[Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977.

[Lewis and Gale, 1994] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. active learning roots.

[Li and Guo, 2013] Li, X. and Guo, Y. (2013). Adaptive active learning for image classification. Technical report.

[Lin and Parikh, 2017] Lin, X. and Parikh, D. (2017). Active learning for visual question answering: An empirical study. *CoRR*, abs/1711.01732.

[Liu et al., 2008] Liu, R., Wang, Y., Baba, T., Masumoto, D., and Nagata, S. (2008). Svm-based active feedback in image retrieval using clustering and unlabeled data. *Pattern Recogn.*, 41(8):2645–2655.

[Luo et al., 2004] Luo, T., Kramer, K., Samson, S., Remsen, A., Goldgof, D., Hall, L., and Hopkins, T. (2004). Active learning to recognize multiple types of plankton. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 478–481 Vol.3.

[Mccallum and Nigam, 1998] Mccallum, A. and Nigam, K. (1998). Employing em in pool-based active learning for text classification.

[Nguyen et al., 2022] Nguyen, V.-L., Shaker, M., and Hüllermeier, E. (2022). How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111.

[Ponzetto, 2021] Ponzetto, P. D. S. (2021). Text analytics lecture 9.

[Qi et al., 2008] Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., and Zhang, H.-J. (2008). Two-dimensional active learning for image classification. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Qu et al., 2020] Qu, Z., Du, J., Cao, Y., Guan, Q., and Zhao, P. (2020). Deep active learning for remote sensing object detection. *CoRR*, abs/2003.08793.

[Reyes et al., 2018] Reyes, O., Altalhi, A. H., and Ventura, S. (2018). Statistical comparisons of active learning strategies over multiple datasets. *Knowledge-Based Systems*, 145:274–288.

[Saito et al., 2015] Saito, P. T., Suzuki, C. T., Gomes, J. F., de Rezende, P. J., and Falcão, A. X. (2015). Robust active learning for the diagnosis of parasites. *Pattern Recogn.*, 48(11):3572–3583.

[Sanh et al., 2019a] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019a). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

[Sanh et al., 2019b] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019b). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

[Sanh et al., 2021] Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Biderman, S., Gao, L., Bers, T., Wolf, T., and Rush, A. M. (2021). Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

[Scheffer et al., 2001] Scheffer, T., Decomain, C., and Wrobel, S. (2001). Active hidden markov models for information extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, IDA '01, page 309–318, Berlin, Heidelberg. Springer-Verlag.

[Schröder et al., 2021] Schröder, C., Niekler, A., and Potthast, M. (2021). Uncertainty-based query strategies for active learning with transformers. *CoRR*, abs/2107.05687.

[Schröder et al., 2021] Schröder, C., Müller, L., Niekler, A., and Potthast, M. (2021). Small-text: Active learning for text classification in python.

[Sechidis et al., 2011] Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 145–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Sener and Savarese, 2018] Sener, O. and Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Settles, 2009] Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

[Settles et al., 2008a] Settles, B., Craven, M., and Ray, S. (2008a). Multiple-instance active learning. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

[Settles et al., 2008b] Settles, B., Craven, M., and Ray, S. (2008b). Multiple-instance active learning. In *In Advances in Neural Information Processing Systems (NIPS*, pages 1289–1296. MIT Press.

[Seung et al., 1992] Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 287–294, New York, NY, USA. Association for Computing Machinery.

[Shannon, 1948] Shannon, C. (1948). Information theory (shannon 1948).

[Shi et al., 2020] Shi, W., Feng, Y., Cheng, G., Liu, S., and Liu, Z. (2020). Multi-category classification problem oriented subsampling-based active learning method. *Journal of Physics: Conference Series*, 1631:012003.

[Tang et al., 2002] Tang, M., Luo, X., and Roukos, S. (2002). Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 120–127, USA. Association for Computational Linguistics.

[Thompson et al., 1999] Thompson, C. A., Califf, M. E., and Mooney, R. J. (1999). Active learning for natural language parsing and information extraction.

[Tuia et al., 2012] Tuia, D., Muñoz Marí, J., and Camps-Valls, G. (2012). Remote sensing image segmentation by active queries. *Pattern Recogn.*, 45(6):2180–2192.

[Tür et al., 2003] Tür, G., Schapire, R. E., and Hakkani-Tür, D. Z. (2003). Active learning for spoken language understanding. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, 1:I–I.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Wang et al., 2017] Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2017). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600.

[Yang et al., 2009] Yang, B., Sun, J.-T., Wang, T., and Chen, Z. (2009). Effective multi-label active learning for text classification.

[Yang et al., 2018] Yang, L., MacEachren, A., Mitra, P., and Onorati, T. (2018). Visually-enabled active deep learning for (geo) text and image classification: A review. *ISPRS International Journal of Geo-Information*, 7:65.

[Yang, 2001] Yang, Y. (2001). A study of thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 137–145, New York, NY, USA. Association for Computing Machinery.

[Yuan et al., 2020] Yuan, M., Lin, H.-T., and Boyd-Graber, J. (2020). Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

[Zhang et al., 2020] Zhang, A., Li, B., Wang, W., Wan, S., and Chen, W. (2020). Mii: A novel text classification model combining deep active learning with bert. *Computers, Materials Continua*, 63:1499–1514.

[Zhang et al., 2008] Zhang, D., Wang, F., Shi, Z., and Zhang, C. (2008). Localized content based image retrieval by multiple instance active learning. *2008 15th IEEE International Conference on Image Processing*, pages 921–924.

[Zhu et al., 2010] Zhu, J., Wang, H., Tsou, B. K., and Ma, M. (2010). Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1323–1331.

[Zhu et al., 2008] Zhu, J., Wang, H., Yao, T., and Tsou, B. (2008). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. pages 1137–1144.

Appendix A

Program Code / Resources

The source code, a documentation, some usage examples, and additional test results are available at the a Google Drive Folder that can be accessed with the following link: https://drive.google.com/drive/folders/1kQ-cLNBmEOvgJp-10ZS–Z0BK4dFqeqO?usp=sharing

A PDF version of this thesis is also emailed to my supervisor.

```
import sys
sys.path.insert(0, 'C:/Users/Emanuela/Documents/Uni/thesis/small-text')
sys.path.insert(0, 'C:/Users/Emanuela/Documents/Uni/thesis/small-text/examples')
```

```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
!pip install transformers
!pip install numpy==1.20.0
!pip install datasets
```

```
import logging

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datasets import load_dataset
from transformers import AutoTokenizer

from small_text.active_learner import PoolBasedActiveLearner
from small_text.initialization import random_initialization_stratified
from small_text.integrations.transformers import TransformerModelArguments
from small_text.integrations.transformers.classifiers.factories import TransformerBasedCla
from small_text.query_strategies import PoolExhaustedException, EmptyPoolException
from small_text.query_strategies import RandomSampling, LeastConfidence

from examplecode.data.example_data_multilabel import get_train_test
from examplecode.data.example_data_transformers import preprocess_data
from examplecode.shared import evaluate_multi_label

#TRANSFORMER_MODEL = TransformerModelArguments('distilbert-base-german-cased')  #distilber
TRANSFORMER_MODEL = TransformerModelArguments('distilroberta-base')
```

Data Preparation

```
eurlex_en = load_dataset('multi_eurlex', 'en')
```

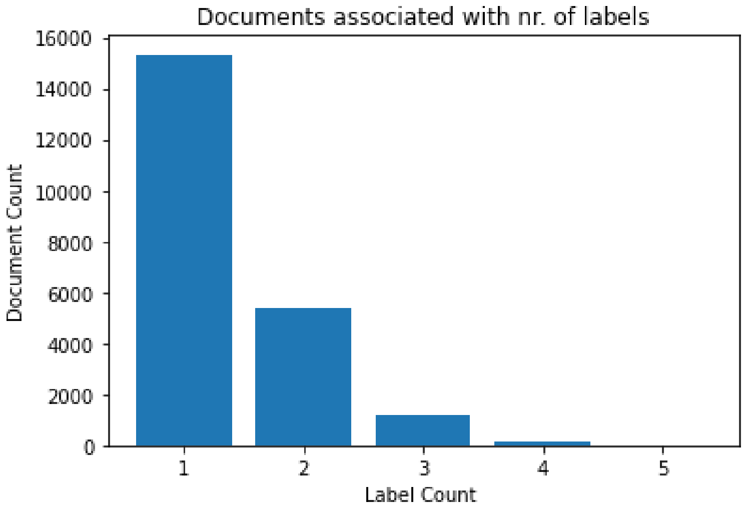
```
train_en=eurlex_en['train']
train_en.features
```

```
test_en = filter_labels(eurlex_en['test'])
val_en = filter_labels(eurlex_en['validation'])
```

```
eurlex_train= pd.DataFrame.from_dict(train_en)
eurlex_test= pd.DataFrame.from_dict(test_en)


def get_unique_labels(df):
    _unique_labels_ = set()

    for labels in list(df["labels"]):
        _unique_labels_.update(labels)
    return _unique_labels_

_train_labels_ = get_unique_labels(eurlex_train)
_test_labels_ = get_unique_labels(eurlex_test)

print("unique topics available,\nTraining dataset len %s\nTesting dataset len %s " % (len(

unavailable_labels = _train_labels_ - _test_labels_
unavailable_labels.update(_test_labels_ - _train_labels_)


## filter unavailable topics in training and testing dataset, as per the above condition
def filter_rows_labels(df,unavailable_labels):
    labels_filter = []
    for label_list in df.labels:#range(len(final_training_df))
        #print(topic_list , unavailable_labels )
        #print(any(topic_list for x in unavailable_labels))
        if any(x in label_list for x in unavailable_labels):
            labels_filter.append(False)
        else:
            labels_filter.append(True)
    return labels_filter



print("Commonly available label len %s" %(len(_train_labels_) - len(unavailable_labels)))

## remove from training dataset
eurlex_train = eurlex_train[filter_rows_labels(eurlex_train,unavailable_labels)]

_train_labels_ = get_unique_labels(eurlex_train)
_test_labels_ = get_unique_labels(eurlex_test)
print("unique topics available,\nTraining dataset len %s\nTesting dataset len %s " % (len(

unavailable_labels = _train_labels_ - _test_labels_
unavailable_labels.update(_test_labels_ - _train_labels_)

eurlex_test = eurlex_test[filter_rows_labels(eurlex_test,unavailable_labels)]
#final_training_df = final_training_df.reset_index()
print(eurlex_train.shape,eurlex_test.shape)


eurlex_train['labels_length'] = eurlex_train["labels"].apply(len)
label_length_df = eurlex_train[["celex_id","labels_length"]].groupby("labels_length").agg(
label_length_df["labels_length"] = label_length_df.index
label_length_df.index = label_length_df["celex_id"]
label_length_df = label_length_df.drop(columns=["celex_id"])
```

```
plt.bar(label_length_df["labels_length"], height= label_length_df.index.tolist())

plt.ylabel('Document Count')
plt.xlabel('Label Count ')
plt.title("Documents associated with nr. of labels")
```

```
Text(0.5, 1.0, 'Documents associated with nr. of labels')
```



```
trainY = eurlex_train["labels"].tolist()
```

```
def perform_active_learning(active_learner, train, labeled_indices, test):
    # Perform 10 iterations of active learning...
    for i in range(10):
        # ...where each iteration consists of labelling 20 samples
        q_indices = active_learner.query(num_samples=100)

        # Simulate user interaction here. Replace this for real-world usage.
        y = train.y[q_indices]

        # Return the labels for the current query to the active learner.
        active_learner.update(y)

        labeled_indices = np.concatenate([q_indices, labeled_indices])

        print('Iteration #{:d} ({} samples)'.format(i, len(labeled_indices)))
        evaluate_multi_label(active_learner, train[labeled_indices], test)


def initialize_active_learner(active_learner, y_train):

    x_indices_initial = random_initialization_stratified(y_train, n_samples=500)
    y_initial = y_train[x_indices_initial]

    active_learner.initialize_data(x_indices_initial, y_initial)

    return x_indices_initial
```

```
def main():
    # Active learning parameters
    num_classes = 21
    clf_factory = TransformerBasedClassificationFactory(TRANSFORMER_MODEL,
                                                        num_classes,
                                                        kwargs=dict({
                                                            'device': 'cuda',
                                                            'multi_label': True
                                                        }))
    query_strategy = LeastConfidence()

    # Prepare some data
    #train, test = get_train_test()
    #train = eurlex_de['train']
    #test = eurlex_de['validation']
    tokenizer = AutoTokenizer.from_pretrained(TRANSFORMER_MODEL.model, cache_dir='.cache/'
    x_train = preprocess_data(tokenizer, train_en['text'], train_en['labels'], multi_label
    x_test = preprocess_data(tokenizer, val_en['text'], val_en['labels'], multi_label=True
    # Active learner
    active_learner = PoolBasedActiveLearner(clf_factory, query_strategy, x_train)


    labeled_indices = initialize_active_learner(active_learner, x_train.y)

    try:
        perform_active_learning(active_learner, x_train, labeled_indices, x_test)
    except PoolExhaustedException:
        print('Error! Not enough samples left to handle the query.')
    except EmptyPoolException:
        print('Error! No more samples left. (Unlabeled pool is empty)')


if __name__ == '__main__':
    logging.getLogger('small_text').setLevel(logging.INFO)

    main()
```

Appendix B

Further Experimental Results

In the following are presented further experimental results.

Table B.1: MulitEurlex EN - DistilRoberta initialized with 1000 samples

Nr. Samples	RS	LC	BT
50	0,65	0,66	0,65
100	0,67	0,67	0,68
150	0,67	0,68	0,66
200	0,68	0,67	0,68
250	0,68	0,68	0,68
300	0,69	0,69	0,7
350	0,70	0,69	0,7
400	0,71	0,71	0,7
450	0,71	0,69	0,71
500	0,71	0,71	0,69
550	0,71	0,73	0,73
600	0,70	0,73	0,71
650	0,72	0,73	0,71
700	0,73	0,74	0,71
750	0,73	0,74	0,7
800	0,74	0,73	0,7
850	0,74	0,73	0,71
900	0,74	0,74	0,71
950	0,74	0,74	0,71
1000	0,74	0,75	0,72

Table B.2: MulitEurlex EN - BERT initialized with 500 samples

Nr.Samples	RS	LC	Kmeans	PE
100	0,62	0,68	0,68	0,65
200	0,67	0,68	0,68	0,67
300	0,68	0,70	0,67	0,70
400	0,70	0,71	0,73	0,71
500	0,71	0,71	0,73	0,71
600	0,73	0,72	0,72	0,72
700	0,72	0,75	0,72	0,72
800	0,74	0,75	0,73	0,73
900	0,73	0,74	0,75	0,74
1000	0,74	0,76	0,74	0,75

Table B.3: Multi Eurlex English - DistilRoberta inialized with 500

Nr. Samples	RS	LC	KM	PE	BT	Best
100	0,66	0,68	0,68	0,66	0,65	0,73
200	0,70	0,70	0,69	0,69	0,69	0,73
300	0,71	0,70	0,70	0,70	0,69	0,73
400	0,70	0,72	0,70	0,71	0,69	0,73
500	0,72	0,73	0,73	0,73	0,70	0,73
600	0,73	0,74	0,74	0,73	0,71	0,73
700	0,73	0,74	0,74	0,74	0,71	0,73
800	0,74	0,75	0,74	0,75	0,72	0,73
900	0,75	0,75	0,74	0,74	0,71	0,73
1000	0,75	0,75	0,75	0,76	0,72	0,73

Table B.4: Multi Eurlex English - DistilRoberta inialized with 1000

Nr. Samples		RS		LC	PE	Kmeans
100		0,72		0,69	0,74	0,72
200		0,72		0,69	0,73	0,71
300		0,74		0,72	0,73	0,73
400		0,73		0,69	0,75	0,75
500		0,73		0,71	0,73	0,74
600		0,75		0,74	0,75	0,74
700		0,75		0,74	0,76	0,74
800		0,75		0,74	0,76	0,75
900		0,74		0,75	0,77	0,75
1000		0,75		0,76	0,77	0,75
1100		0,75		0,76	0,77	0,75
1200		0,76		0,76	0,77	0,76
1300		0,76		0,77	0,77	0,76
1400		0,76		0,77	0,77	0,75
1500		0,76		0,78	0,77	0,76
1600		0,76		0,78	0,77	0,75
1700		0,76		0,77	0,77	0,75

Table B.5: Multi Eurlex English - DistilBERT inialized with 500

Nr. Samples	RS	LC	PE	BT	Best
100	0,64	0,66	0,66	0,64	0,73
200	0,65	0,69	0,69	0,67	0,73
300	0,68	0,70	0,69	0,67	0,73
400	0,67	0,70	0,69	0,68	0,73
500	0,67	0,72	0,70	0,69	0,73
600	0,70	0,74	0,72	0,69	0,73
700	0,71	0,75	0,74	0,70	0,73
800	0,72	0,75	0,72	0,72	0,73
900	0,73	0,75	0,74	0,72	0,73
1000	0,73	0,75	0,75	0,71	0,73

Table B.6: GoEmotions- DistilRoberta

Nr. Samples	LC	KM	BT	PE
100	0,40	0,32	0,37	0,40
200	0,42	0,38	0,40	0,43
300	0,43	0,40	0,41	0,41
400	0,45	0,40	0,43	0,44
500	0,45	0,43	0,44	0,44
600	0,46	0,43	0,44	0,46
700	0,47	0,44	0,46	0,45
800	0,47	0,45	0,45	0,48
900	0,48	0,46	0,45	0,48
1000	0,49	0,47	0,49	0,48

Table B.7: Emotions- DistilBert

Nr. Samples	RS	LC	PE	BT
100	0,33	0,36	0,38	0,39
200	0,37	0,40	0,39	0,39
300	0,41	0,41	0,39	0,43
400	0,41	0,42	0,42	0,42
500	0,43	0,43	0,43	0,42
600	0,45	0,45	0,45	0,46
700	0,44	0,43	0,44	0,44
800	0,44	0,44	0,45	0,45
900	0,43	0,44	0,46	0,45
1000	0,43	0,45	0,44	0,47

Table B.8: MiniEurlex-DistilBert

Nr. Samples	RS	LC	PE	BT
100	0,80	0,77	0,76	0,78
200	0,79	0,78	0,79	0,78
300	0,80	0,77	0,74	0,78
400	0,79	0,80	0,77	0,81
500	0,82	0,80	0,81	0,80
600	0,82	0,79	0,81	0,79
700	0,81	0,79	0,79	0,80
800	0,81	0,80	0,79	0,78
900	0,80	0,79	0,80	0,80
1000	0,82	0,81	0,80	0,80

Table B.9: MiniEurlex-DistilRoberta

Nr. Samples	RS	LC	PE	BT
100	0,78	0,81	0,76	0,78
200	0,80	0,79	0,79	0,79
300	0,82	0,80	0,78	0,79
400	0,82	0,81	0,82	0,78
500	0,83	0,81	0,78	0,81
600	0,82	0,81	0,81	0,80
700	0,82	0,81	0,82	0,82
800	0,83	0,79	0,80	0,81
900	0,82	0,81	0,83	0,81
1000	0,83	0,81	0,82	0,81

Table B.10: MultiEurlex Spanish- Distilbert-espanol

Nr. Samples	RS	LC	PE	BT
100	0,69	0,69	0,69	0,69
200	0,70	0,72	0,70	0,70
300	0,73	0,72	0,71	0,72
400	0,69	0,74	0,73	0,70
500	0,74	0,75	0,73	0,71
600	0,74	0,76	0,73	0,72
700	0,73	0,76	0,74	0,71
800	0,76	0,76	0,76	0,71
900	0,73	0,77	0,76	0,73
1000	0,76	0,77	0,75	0,73

Table B.12: DistilRoberta Eurlex EN Initialized with 1000 +25*20

Nr. Samples		**RS**	**LC**	**BT**
50		0,65	0,66	0,65
100		0,67	0,67	0,68
150		0,67	0,68	0,66
200		0,68	0,67	0,68
250		0,68	0,68	0,68
300		0,69	0,69	0,7
350		0,70	0,69	0,7
400		0,71	0,71	0,7
450		0,71	0,69	0,71
500		0,71	0,71	0,69
550		0,71	0,73	0,73
600		0,70	0,73	0,71
650		0,72	0,73	0,71
700		0,73	0,74	0,71
750		0,73	0,74	0,7
800		0,74	0,73	0,7
850		0,74	0,73	0,71
900		0,74	0,74	0,71
950		0,74	0,74	0,71
1000		0,74	0,75	0,72

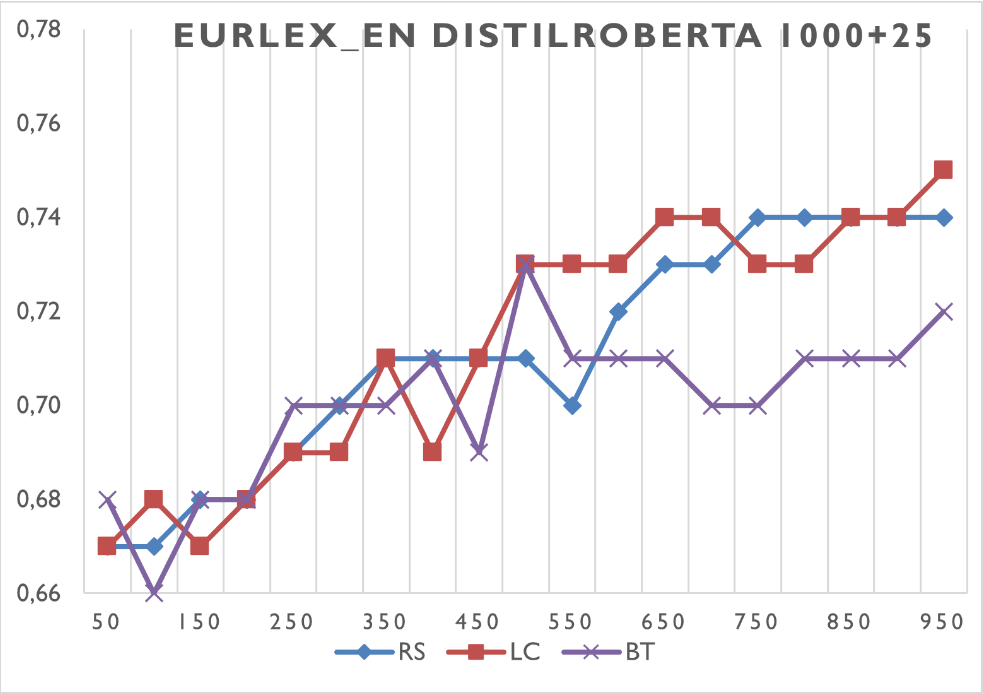


Figure B.1: Performance of DistilRoberta on Eurlex over 20 iterations, with a sample size of 25

Appendix C

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Masterarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, den 10.03.2022 Unterschrift