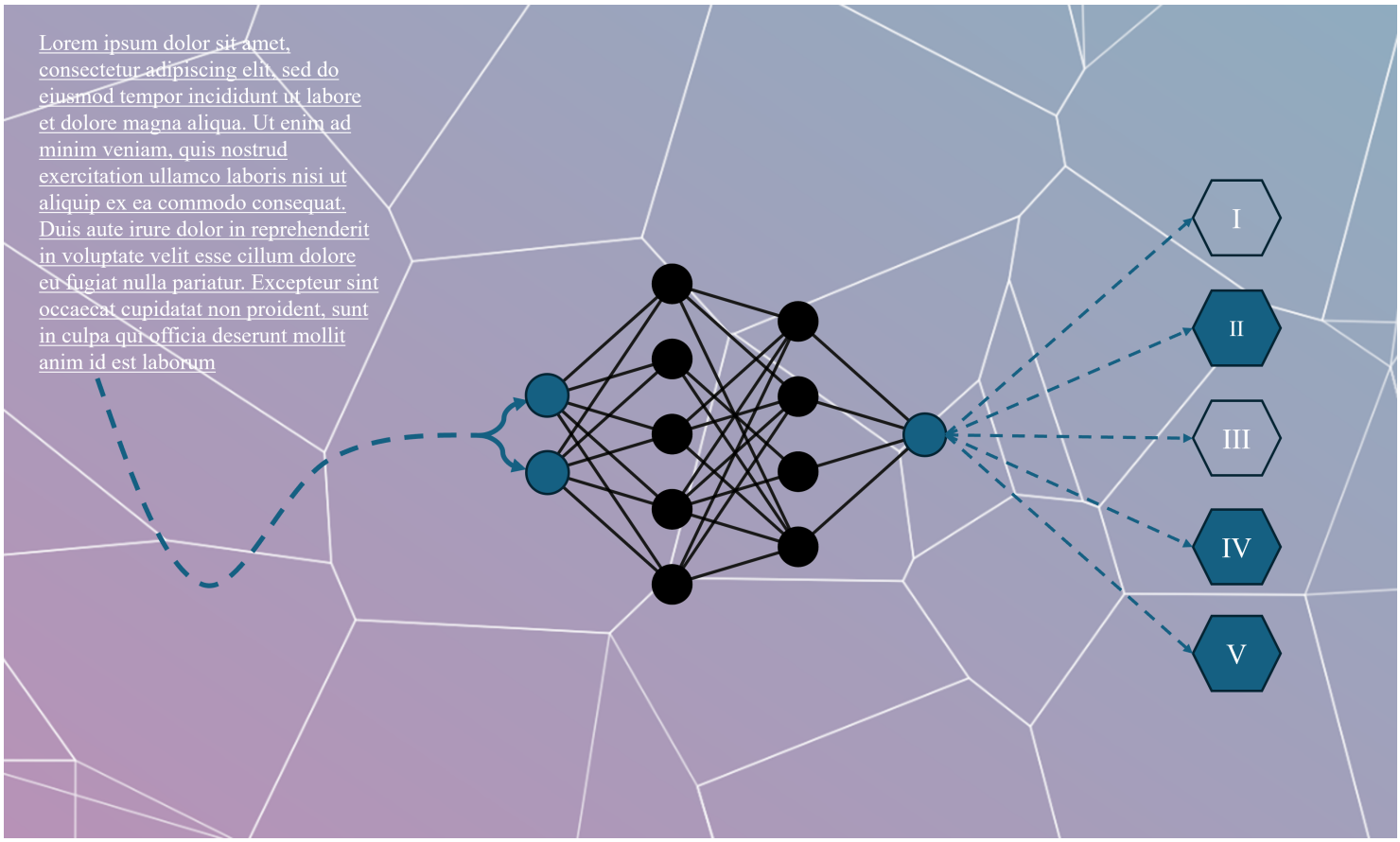
CHALMERS



Applying Conformal Prediction for LLM Multi-Label Text Classification

Application of Conformal Prediction for Robust Large-Language-Model Based Multi-Label Text Classification

Master's thesis in Complex Adaptive Systems

VIKTOR ÖRNBRATT

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2025

www.chalmers.se

MASTERS'S THESIS 2025

Applying Conformal Prediction for LLM Multi-Label Text Classification

Application of Conformal Prediction for Robust Large-Language-Model Based Multi-Label Text Classification

VIKTOR ÖRNBRATT

CHALMERS

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg 2025

Applying Conformal Prediction for LLM Multi-Label Text Classification
Application of Conformal Prediction for Robust
Large-Language-Model Based Multi-label Text Classification
VIKTOR ÖRNBRATT

Supervisor: Johan Hallberg Szabadváry, Algorithma AB
Examiner: Giovanni Volpe, DEPARTMENT OF PHYSICS

MASTER'S THESIS 2025
DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
SE-412 96 Gothenburg
Phone +46 31 772 1000

Front Cover: A flowchart showcasing a text being classified by minimalist model of a neural net into multiple labels with a background created via a Voronoi diagram.

Typeset in LaTeX
Göteborg 2025

Applying Conformal Prediction for LLM Multi-Label Text Classification
Application of Conformal Prediction for Robust
Large-Language-Model Based Multi-label Text Classification
Viktor Örnbratt
Department of Physics
Chalmers University of Technology

Abstract

This thesis investigates how conformal prediction can be used to improve the robustness and interpretability of multi-label text classification with large language models (LLMs). Using a dataset of Wikipedia comments annotated for multiple types of toxicity, a binary relevance approach is combined with inductive conformal prediction to produce label-wise prediction sets with formal coverage guarantees.
Two data splitting strategies are explored to study the trade-off between model accuracy and calibration quality: one prioritising LLM fine-tuning, the other prioritising calibration set size.
Results show that conformal prediction enables meaningful uncertainty quantification, including abstention on ambiguous inputs, while maintaining reliable coverage across a range of significance levels. The analysis also highlights challenges related to rare labels, label imbalance, and the sensitivity of validity guarantees to shifts in annotation quality and dataset distribution over time.
Overall, the study supports the practical use of conformal prediction as a safeguard mechanism for LLM-based classifiers, especially in settings where predictive reliability and human oversight are both critical.

Keywords: Large Language Models, Conformal Prediction, Multi-label Conformal Prediction, Uncertainty Quantification, Text Classification

Acknowledgements

I would first like to express my appreciation to my supervisor Johan Hallberg and his supervision throughout the thesis. His introduction to the field of conformal prediction has both been both interesting and deeply appreciated. Additionally I would also like to extend my appreciation to everyone at Algorithma AB as they provided me with tons of energy with their great vibes and insightful conversations during the course of this work.
Finally I would like to give a huge thanks to both my friends and family for their unwavering support throughout the years at Chalmers. Without it my time at Chalmers could never have been as wonderful as it has been.

Viktor Örnbratt, Gothenburg, June 2025

Abbreviations

Below is the list of abbreviations that have been used throughout this thesis listed in alphabetical order:

ICP	Inductive Conformal Prediction
LLM	Large Language Model
MLCP	Multi-label Conformal Prediction
NLP	Natural Language Processing
TCP	Transductive Conformal Prediction
UQ	Uncertainty Quantification

Contents

List of Figures

List of Tables

1

Introduction

Large language models (LLMs) have rapidly grown in use since 2022, with ChatGPT, Llama and Claude being some of the more popular models. One issue that has become prominent within LLMs is the case of hallucinations. A hallucination occurs when an LLM generates an output that is irrelevant to the prompted question, completely or partially untrue, or inconsistent with the input data. In classification tasks where hallucination in the generative sense does not directly apply, the corresponding concerns manifest as misclassifications or classification uncertainty. To address these issues in case of text classification with LLMs, conformal prediction has been considered a suitable method for evaluating predictions against the model's past accuracy.

1.1 Background

Although LLMs have proven themselves to be powerful, there has also been a wave of examples of their unreliability and uncertainty. One way to partially account for this problem in LLM text classification is through a method called conformal prediction.

Conformal prediction is a way to estimate uncertainty by outputting prediction intervals instead of single-point predictions. This is done by first calculating nonconformity scores for all previously labelled data whereafter these scores are used to create prediction intervals for unlabelled test data. The prediction intervals in classification are represented by a set with classes and model prediction probability for the classes above a pre-specified threshold.

It is important to note that the type of conformal prediction that will be used is the offline variant. This is because the online variant requires the predictor to be retrained between each calculation of conformity measure for class to class prediction. When looking at LLMs the amount of compute that retraining would require would be too costly for any application of the method.

As for the background for why Algorithma AB wants the thesis to be carried out, Algorithma AB is an AI and algorithmic consulting company based in Gothenburg and Stockholm in Sweden. The company's interest within this work is to explore

LLM multi-label text classification for research purposes as well to assess the possibility of this work being integrated into future tailor-made solutions for possible customers of the company.

1.2 Societal, Ethical and Ecological Aspects of the Thesis

The thesis will also discuss possible societal, ethical and ecological implications and concerns that can stem from the results and application of the methods used in the thesis.
For example, regarding the ecological aspects the amount of compute required for LLMs is considerable and therefore the emissions from the energy sources must be assessed to evaluate the environmental impact. This energy cost could be partially mitigated by either smaller more specifically trained models or by model distillation. From an ethical perspective, it is important to note that while conformal prediction can measure the uncertainty within a prediction from an LLM, the underlying model is in most cases a black-box and can therefore contain a range of biases from the training data. This limitation the method cannot be fully corrected for and therefore care has to be taken when implementing it to ensure fairness in the predictions.

1.3 Purpose and Research Question

The purpose of this thesis is to investigate the application of conformal prediction on text multi-label classification performed by LLMs. The specific conformal prediction method that will be used to evaluate the LLMs classification effectiveness will be inductive conformal prediction (ICP), in addition to exploring the variations within multi-label classification. In the event that the report is fully successful at estimating the uncertainty of LLM predictions the results may be further developed and potentially commercialized as part of an AI solution to potential customers of Algorithma AB. This leads to the following research question for the thesis:

How can inductive conformal prediction be applied to large language models in multi-label text classification tasks to produce reliable uncertainty estimates?

1.4 Limitations of the Thesis

The thesis won't treat the application of "online" or transductive conformal prediction due to limits of the computational cost of the method combined with the LLM training.
Additionally, due to the large number of hyperparameters involved in testing and optimizing LLMs the scope of models considered was limited as otherwise the experiments would quickly become infeasible due computational cost and time constraints.

1.5 Use of AI within the Thesis

AI has been used to help generate code for plotting figures and early drafts of figures such as Figures A.1 and 2.1. Additionally, it has been used to assist with proofreading, and to suggest expansion of sections within the thesis. No sources were directly taken from AI content and any information gained while drafting text was thoroughly double-checked before use and citation within the thesis. The choice of AI use in this way was to improve the workflow and increase the energy allotted to critical decisions and results from the thesis.

2

Theory

In the following sections, there will be an overview of large language models, classification problems and the uncertainty quantification framework that is conformal prediction.

2.1 Large Language Models

Large language models (LLMs) are transformer-based architectures pre-trained on vast amounts of text data, enabling them to capture complex language patterns and contextual information. They have demonstrated exceptional performance on a wide range of Natural Language Processing (NLP) tasks, including sentiment analysis, question answering, and text classification. Their scalability and adaptability make them a cornerstone of almost all modern NLP applications.

2.1.1 Natural Language Processing

Natural Language Processing (NLP) is a subdiscipline of artificial intelligence focused on interactions between computers and human languages. The aim of NLP is to allow machines to interpret, understand and generate human language in a meaningful and useful way. This in turn has several applications including but not limited to translation, sentiment analysis, and text classification [1].
The earliest iterations of NLP relied heavily on rule-based systems and statistical methods. Including approaches such as bag-of-words model, n-grams and shallow parsing i.e. dividing the sentences into meaningful phrases. [1]
Later, in the year 2013 and 2014 came the introduction of word embeddings such as Word2Vec [2] and GloVe [3] which enabled models to represent words in continuous vector spaces that captured semantic similarity. This together with Recurrent Neural Networks and Long Short-Term Memory networks that helped the models account for temporal dependencies within text. This led to a high degree of success but the models still had issues with handling both long-range dependencies and parallelization.
This missing piece came with the architecture introduced in the paper “Attention Is All You Need“ [4]. The transformers architecture replaced the recurrence with

self-attention, leading to significant improvements in both training efficiency and model performance. The transfomers architecture laid the foundation for modern LLMs such as BERT [5], ROBERTA [6], ELECTRA [7] and GPT [8].
As LLM development continues rapidly their ability to generalize across different tasks has made them the obvious choice in NLP pipelines including that of Multi-label text classification used in this thesis.

2.1.2 Pre-trained Language Models

Pre-trained LLMs have grown in popularity in recent years with providers such as Ollama and Hugging Face offering libraries of different pre-trained models. These models are now often either deployed directly for a task or further adjusted by fine-tuning on specific datasets. In this thesis, three pre-trained models were primarily considered for the multi-label text classification using the Transformer package maintained by both the online community and the company Hugging Face. These models being BERT introduced in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [5], ROBERTA: A Robustly optimized BERT pretraining Approach [6] and ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators [7].

BERT

The BERT model uses only the encoder portion of the Transformer architecture. Where each block consists of two main subcomponents: Multi-head self-attention and feed-forward networks. Multi-head self-attention lets each token account for every other token in the given input sequence thereby allowing the model to capture long-range dependencies. The second component, feed-forward networks, transforms the outputs of the attention heads into more informative representations. Unlike traditional language models that process text unidirectionally, BERT uses a bidirectional transformer encoder, allowing it to jointly condition on both the left and right context of each word. This significantly increases its contextual understanding and thereby its performance of a variety of NLP tasks including classification [5]. Relevant to this thesis is that the data is both BERT, RoBERTa, and ELECTRA are all trained on Wikipedia and BooksCorpus. This is relevant as the example dataset used in the method is a open-source wikipedia comment dataset [9].

RoBERTa

The RoBERTa model is a robustly optimized variant of the BERT architecture. Like BERT, RoBERTa uses only the encoder portion of the Transformer, where each block consists of two main subcomponents: multi-head self-attention and feed-forward networks. Multi-head self-attention enables each token to attend to every

other token in the input sequence, capturing complex dependencies within the text. The feed-forward networks then transform the outputs of the attention heads into richer representations for downstream tasks.
RoBERTa improves upon BERT by making key modifications in its pretraining procedure. Specifically, it removes the Next Sentence Prediction (NSP) objective, trains on larger mini-batches, uses a larger corpus, and increases the number of training steps, allowing the model to learn more robust contextual representations. Like BERT, RoBERTa uses a bidirectional encoder, jointly conditioning on both the left and right context of each word, which enhances its understanding of natural language. This makes it particularly effective for classification tasks [6].

ELECTRA

The ELECTRA model also builds upon the Transformer encoder architecture, consisting of multiple layers each composed of multi-head self-attention and feed-forward networks. As with BERT and RoBERTa, multi-head self-attention in ELECTRA enables each token to incorporate information from every other token in the input sequence, capturing rich contextual dependencies. The feed-forward networks then refine these representations for downstream tasks.
ELECTRA introduces a novel pretraining objective known as the replaced token detection task, which differs fundamentally from the masked language modeling objective used by BERT and RoBERTa. Instead of masking tokens and predicting them, ELECTRA trains a discriminator to distinguish between real and replaced tokens generated by a small generator model. This adversarial-like setup allows ELECTRA to learn more efficiently from unlabeled data and typically requires less compute to reach comparable performance[7].
Although ELECTRA's pretraining is also performed on large text datasets like Wikipedia and BooksCorpus, it leverages this replaced token detection task to achieve faster convergence and improved performance on classification benchmarks.

2.1.3 AutoModelForSequenceClassification

In practice, many NLP tasks can be addressed using pre-trained models available through libraries like Hugging Face's Transformers. One of the most versatile classes is AutoModelForSequenceClassification which is used in this thesis. It automatically selects the appropriate architecture for text classification tasks based on the specified checkpoint, though further alterations are often needed to adjust it to its specific task. This abstraction greatly simplifies model selection and iteration, as it automatically loads the corresponding backbone (e.g., BERT, RoBERTa) and adds a classification head suited to the number of classes/labels in the target task.
When using AutoModelForSequenceClassification, the model outputs logits (unnor-

malized scores for each class/label). Applying the appropriate activation function (softmax for multi-class, sigmoid for multi-label) converts these logits into probability scores, which can be used within both model training and conformal prediction [10].

2.2 Classification Problems

Classification problems can be divided into three different types: binary classification (predicting one of two possible classes), multi-class classification (predicting one of three or more classes) or multi-label classification (predicting one or more labels out of three or more possible labels). In the order presented, each variation of problem gains additional intricacies, with multi-label being the most complex of these three presented.

2.2.1 Supervised Learning

A core category of machine learning is supervised learning, where models are trained on labelled data to learn a way for mapping inputs to outputs. In classification problems, the goal is to learn a function that can accurately predict these labels y given an input x. The learning process in this case of the models is achieved by minimizing a loss function that measures the difference between the models predicted labels and the ground-truth labels [11].

2.2.2 Multi-class vs Multi-label Classification

For multi-class classification problems, cross entropy loss is frequently used. It measures the difference between the predicted class probability distribution and the true class label. The function can be defined as:

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y_k \log \hat{p}_k \tag{2.1}$$

where y_k is 1 for the true class and 0 for all other classes and the models predicted probability distribution is $\hat{p} = (\hat{p}_1, ..., \hat{p}_K)$ over the K number of classes [12].
In multi-label classification, where each instance can belong to multiple classes simultaneously, the output is typically modelled as a vector of independent binary labels. That means that each label y_ℓ is treated as a separate binary classification task. In this case, the loss function used is binary cross entropy (BCE) is used:

$$\mathcal{L}_{BCE} = -\frac{1}{L}\sum_{\ell=1}^{L}[y_\ell \log \hat{p}_\ell + (1 - y_\ell) \log 1 - \hat{p}_\ell] \tag{2.2}$$

where p_ℓ is the models predicted probability for each label L. Binary cross entropy treats each label independently, which is well suited to multi-label problems where labels may have independent distributions [13].

2.2.3 Evaluation Metrics for Multi-label Classification

For evaluation of the of the LLMs predictive performance we use two types of metrics, subset accuracy and label-wise recall. Subset accuracy in the multi-label case is a strict metric as it requires a complete match between the true vector of label and the predicted for each instance. The label-wise accuracy, by contrast, is less strict and provides insight into which labels are hardest to predict. In the below equations I represents the binary indicator function, $\hat{y}$ the predicted labels, y the true labels. i being the index iterating across all instances and j iterating across all instance containing label ℓ [14].

$$Acc_s = \frac{1}{N}\sum_{i=1}^{N} I[\hat{y}_i = y_i] \tag{2.3}$$

$$Acc_\ell = \frac{1}{M}\sum_{j=1}^{M} I[\hat{y}_j^\ell = y_j^\ell] \tag{2.4}$$

2.3 Conformal Prediction & Uncertainty Theory

As machine learning models are increasingly deployed in real-world applications, it becomes increasingly important to not only to make accurate predictions but also to understand the reliability of these predictions. Conformal Prediction is a powerful, model-agnostic, and distribution-free framework that allows the quantification of uncertainty by producing prediction sets with a guaranteed error rate under minimal assumptions.

2.3.1 Introduction to Uncertainty Quantification

Uncertainty in the context of machine learning refers to the degree of confidence or trust one can place in a model's prediction. In Hüllermeier and Waegeman (2021) [15] present two types of uncertainty: aleatoric and epistemic.
Aleatoric uncertainty arises from inherent noise in the data. For instance, in text classification, the same sentence may be labelled differently by different annotators due to ambiguity or subjectivity. This type of uncertainty cannot be reduced by collecting more data, as it reflects variability in the data-generating process itself.
Epistemic uncertainty on the other hand is related to the model's knowledge about the data. It captures the uncertainty due to limited or incomplete training data, and

it tends to be higher in regions of the input space that the model has not seen during training. This type of uncertainty can often be reduced by either incorporating more diverse training data or by refining the model through hyper parameter optimization. Accounting for both of these types of uncertainty is crucial when developing robust machine learning systems [15]. Specifically in NLP pipelines where LLMs that are deployed for multi-label classification may encounter new text-inputs that are ambiguous or out-of-distribution where the model uncertainty is high. While modern neural networks often have issues with hallucinations which leads them to output overconfident predictions in classification pipelines [16]. The conformal prediction framework can help to mitigate this by providing formal guarantees on prediction reliability.

2.3.2 Intuitive Explanation of Conformal Prediction

One of the main topics of this thesis is conformal prediction, an uncertainty quantification framework for predictive models. Before formally introducing the framework, is it helpful to get an intuitive mental picture of the how it works. Although the details of this framework is rather intricate, it is quite straightforward to draw a parallel to something you would do in real life in the simplest case of conformal prediction.

Imagine for example the case of predicting the temperature outside based on looking out the window in the morning. Every time we predict this and dress accordingly we can then note down how odd the actual temperature was that day based on the view earlier in the morning. Here an example of rainy weather and very high temperatures could be considered very odd while the case of rainy weather and cold temperature being not so odd. Doing this over the course of a few months we can then get an estimate for both how good our "gut feeling" or model is and what examples is easier or harder to predict. That is the basis of conformal prediction and the calibration that is mentioned later in this thesis.

2.3.3 Fundamentals of Inductive Conformal Prediction

Conformal prediction can be applied to both regression and classification problems. The output of the framework in the regression case produces is a prediction interval where the true point prediction will fall with a specified error rate. In the classification case, which is the focus of this thesis, the output is a predictions set. The prediction set is a set of labels guaranteed to include the true label with a specified confidence level $1 - \epsilon$, where $\epsilon \in (0,1)$ is the chosen significance level.

The conformal prediction framework wraps around traditional machine learning models with a method that accounts for the reliability of their predictions. The framework doesn't modify the underlying model but instead calibrates its output

using a statistical technique based on minimal assumptions [17].

Exchangeability and Validity

A central assumption for conformal prediction is that that the underlying data points for training and calibration are exchangeable. Exchangeability is a relaxed version of the more common i.i.d. (independent and identically distributed) assumption within statistics. Formally, a finite sequence of examples $z_1, ..., z_n$ is exchangeable if for any permutation of π of the sequences indices {1,...,n}. In other words it means that the joint joint probability distribution of a sequence of random data points remains unchanged for any permutation.

$$Pr(z_1, ..., z_n) = Pr(z_{\pi(1)}, ..., z_{\pi(n)})$$

This assumption ensures that that the ordering of the data has no effect on the amount of information it conveys [18]. It is important because it supports the theoretical guarantee of validity within conformal prediction. This validity guarantee being that in the long-run the rate of that which the true label is excluded for the predictions set does not exceed the chosen significance level ϵ.

The Nonconformity Measure and Prediction Sets

A core part of conformal prediction is the concept of nonconformity. A nonconformity measure is a function that quantifies how "surprising" or "nonconforming" an example ($z = (x,y)$) is relatively to a set of observed examples. Common nonconformity scores for LLM classification include inverse class probabilities using the models logit outputs together with either a softmax or sigmoid activation function. It is worth noting that there are also non-logit based approaches handling prediction frequency, normalized entropy, and semantic similarity as shown by [19] With that in mind, conformal prediction works as follows: when given a new test example, it considers all possible candidate labels ℓ and evaluates how conforming each is by calculating a p-value:

$$p_\ell = \frac{|\{i = 1, ..., n : \alpha_i \geq \alpha_{n+1}\}| + 1}{n + 1} \tag{2.5}$$

where α_i is the nonconformity scores of the i-th calibration example and α_{n+1} is the corresponding score for the test example with candidate label ℓ. This creates the prediction set by collecting all labels for which $p_\ell > \epsilon$ thus ensuring valid coverage. A more detailed explanation regarding how the nonconformity scores were constructed in this thesis can be read in Section 3.4

Inductive Conformal Prediction (ICP)

Although the original transductive version of conformal prediction provides stronger guarantees of exact validity, it requires re-training or re-evaluation of the model for each new test instance, which quickly becomes computationally costly for some models. This means that when working with large machine learning models especially LLMs, ICP is the obvious choice due to its greater computational efficiency. Formally, in ICP the data is split into two distinct subsets: a proper training set used to train the underlying model and a calibration set used to compute nonconformity scores.

Following the definition of Inductive Conformal Predictors given in Subsection 4.2.2: Inductive Conformal Predictors in the Offline and Semi-Online Modes in Algorithmic learning for a Random World 2nd ed [17].

Definition 1. *Inductive Conformal Predictor*

The training set of size c is first split into two parts: the proper training set Z_{train} : $z_1, ..., z_m$ of size $m < c$ and the calibration set of size Z_{cal} : z_{m+1},...z_c of size $c - m$. For every test object $x_i, i = c + 1, ..., c + k$, compute the prediction sets

$$\Gamma^{\epsilon}(z_1, ..., z_c, x_i) := \left\{ y \in \mathcal{Y} : \frac{|\{j = m + 1, ..., c : \alpha_j \geq \alpha_i\}| + 1}{c - m + 1} > \epsilon \right\} \tag{2.6}$$

where the nonconformity scores are defined by

$$\alpha_j := A((z_1, ..., z_m), z_j), \qquad j = m + 1, ..., c$$

$$\alpha_i := A((z_1, ..., z_m), (x_i,y)),$$

Prediction Thresholds

Aside from directly calculating the p-values shown in Equations (2.5) and (2.6) we can define a prediction threshold τ for the nonconformity scores to compare against for label acceptance or rejection. Assuming the nonconformity scores are ordered $\alpha_1 < ... < \alpha_c$ then the predicted label y_ℓ is accepted if $\alpha_i^\ell \leq \alpha_{\lceil c(1-\epsilon) \rceil} = \tau$.

Accounting for calibration set size

An extremely important aspect of both transductive and inductive conformal prediction is the quantity of calibration data. For the coverage guarantees to be valid, they need to be adjusted according to the size of the calibration set size, otherwise there is a risk of overconfident error rates. This adjustment can be done with the following equation:

$$E := \epsilon + \sqrt{\frac{\ln \frac{1}{\delta}}{2h}} \tag{2.7}$$

where if Γ is a conformal predictor, it is (E,δ)-valid, with E described by Equation (2.7) as a joint packed probability together with a pre-defined delta and h being the size of the calibration set used for the adjustment [17].

2.3.4 Multi-label Conformal Prediction

As mentioned previously in section 2.2.2 there is a significant increase in complexity between the multi-class and multi-label case of classification problems. Fortunately, previous research has explored how to tackle these problems with the conformal prediction framework. Currently Multi-label Conformal Prediction (MLCP) has three main approaches: Label Power Set (LPS), Binary Relevance (BR) and Instance reproduction (IR).

The first method, LPS, was disregarded in this thesis due to the issue of label scaling, as the number of classes is defined as $n_{classes} = L!$, where L is the number of labels. This limitation is noted in later work on Efficient LP-MLCP [20]. Additionally the output of LPS is a set of possible label-sets which requires a level of post-processing that is not relevant to the main focus of this thesis. This thesis focuses instead on the second variant: BR-ML approach which divides the multi-label problem into L number of binary conformal classification problems. Here we also get ℓ number of conformal predictors that all need to have their significance level calibrated against the number of label instances for their respective labels.

$$[\Gamma_1^{E_1}, ..., \Gamma_\ell^{E_\ell}] \tag{2.8}$$

The third approach, instance reproduction, is very similar as it also builds upon the binary relevance assumption but instead of creating ℓ number of conformal predictors it instead has one predictor looking at ℓ number of instance label pairs that it is calibrated over. A more in depth comparison and explanation of these different approaches can be found in "A Comparison of Three Implementations of Multi-Label Conformal Prediction" [21].

2.3.5 Conformal Prediction with LLMs

In large language models (LLMs), text classification tasks often require the model to produce class membership probabilities. The transformation of model outputs (logits) into probabilities is essential for applying conformal prediction, which relies on valid measures of uncertainty to construct its prediction sets.

Softmax Activation (Multi-Class Classification)

For multi-class classification tasks, where exactly one class must be assigned, LLMs, typically apply the softmax function to the logits:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{2.9}$$

where z_i is the logit for class i, and K is the number of classes. This activation projects the logits onto a probability simplex, represented geometrically as a triangle in three dimensions (see left side of Figure 2.1). The outputs sum to one, ensuring that the model assigns all probability mass to exactly one class. In this scenario, the model's prediction corresponds to the class with the highest softmax score.

Sigmoid Activation (Multi-Label Classification)

For multi-label tasks, where each label is independent, LLMs instead have to apply a sigmoid activation to each logit individually:

$$\sigma(z_i) = \frac{e^{z_i}}{e^{z_i} + 1} \tag{2.10}$$

This projects each label output to the unit interval [0,1] without enforcing any mutual exclusivity (right side of Figure 2.1). Each dimension is independent, and the output space forms a unit cube in L-dimensional space. This is especially important in tasks like multi-label classification, where multiple classes can be relevant simultaneously.
In multi-label classification, nonconformity scores based on softmax outputs become suboptimal because the probabilities for each label compete with each other. Instead, sigmoid activations are more appropriate, as they allow class probabilities to be treated independently.

Implications for Conformal Prediction

The choice of activation function influences how conformal predictors are designed. In multi-class problems with softmax outputs, conformal predictors typically rely on the top-1 predicted probability or nonconformity scores derived from softmax outputs to generate prediction sets that guarantee a certain coverage. For multi-label problems with sigmoid outputs, each label prediction can be treated independently, allowing the application of binary conformal predictors per label.
Figure 2.1 below illustrates these fundamental differences. The left side shows the 2-simplex formed by softmax outputs, highlighting the constraint that all class probabilities must sum to one. In contrast, the right side shows the unit cube formed by sigmoid outputs, where each class can be assigned a probability independently.

This visualization and distinction between the softmax and sigmoid activations and their geometric interpretations helps us build an intuitive understanding of the increase of complexity between the difference classification problems. Additionally it aids in the explanation of why the multi-label conformal methods in Subsection 2.3.4 are constructed in the they way that they are.

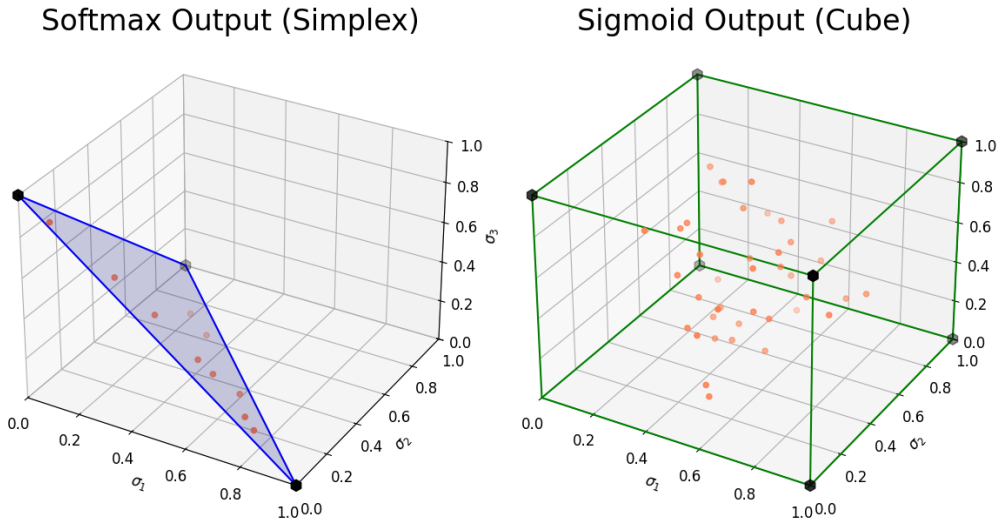


Figure 2.1: Output space for softmax and sigmoid values for the three class and three label cases respectively. With Orange circles showcasing random logits with the activation functions applied and the black hexagons representing the possible ground-truth locations

2.3.6 Evaluation Metrics for MLCP

Looking at conformal prediction, we also use coverage (how often the true label set is a subset of the predicted label set), efficiency (average size of the prediction label set) [17] L-A-efficiency (label averaged efficiency), Jaccard index (for set similarity) and Hamming loss (for average per-label instance prediction errors) [22]. In the below equations I represents the indicator function.

$$Cov_{ML} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \subseteq \Gamma(x_i)) \tag{2.11}$$

$$Effiency = \frac{1}{N}\sum_{i=1}^{N} |\Gamma(x_i)| \tag{2.12}$$

$$\text{L-A-}Effiency = \frac{1}{N \cdot L}\sum_{i=1}^{N} |\Gamma(x_i)| \tag{2.13}$$

$$HL = \frac{1}{N \cdot L}\sum_{i=1}^{N}\sum_{\ell=1}^{L} I(\hat{y}_i^{\ell} \neq y_i^{\ell}) \tag{2.14}$$

$$J_{acc} = \frac{|A \cap B|}{|A \cup B|} \tag{2.15}$$

3

Method

This chapter outlines the methodology employed in this thesis, including data preprocessing, model selection, the conformal prediction pipeline, and prediction set handling.

3.1 Data Overview

The primary dataset used in this thesis is the "Exploring the Toxicity of Wikipedia Comments" dataset available at [9]. This dataset comprises 159,571 Wikipedia comments labeled as either non-toxic or toxic with one or more of six different toxicity labels. The dataset contains 15,294 toxic comments and 144,277 non-toxic comments, showcasing that it is significantly imbalanced with an approximately 9 to 1 ratio of non-toxic to toxic comments.
Each data instance includes an identifier, the comment text, and binary-encoded annotations for the six toxicity labels: *toxic*, *severe toxic*, *insult*, *obscene*, *threat*, and *identity hate*. For this thesis, the dataset was further modified by iterating over all comments and adding an additional *non-toxic* label (with a value of 1) to instances where the sum of the toxic labels was zero.

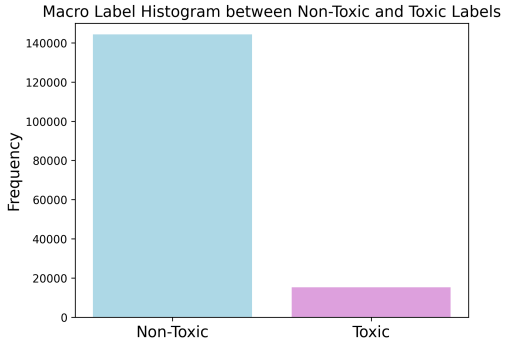


Figure 3.1: Frequency of non-toxic vs all types of toxicity across entire dataset

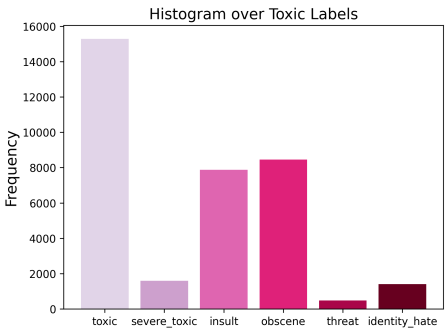


Figure 3.2: Toxic-label frequency across entire dataset

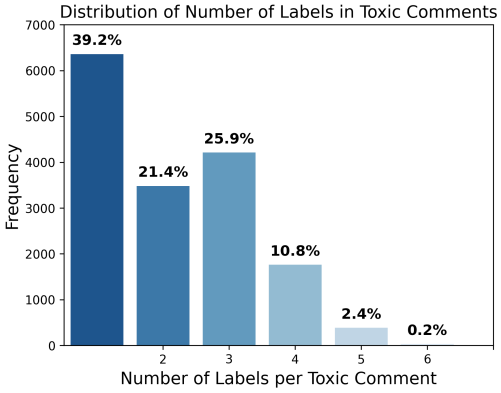


Figure 3.3: Histogram over number of labels per instance for toxic comments.

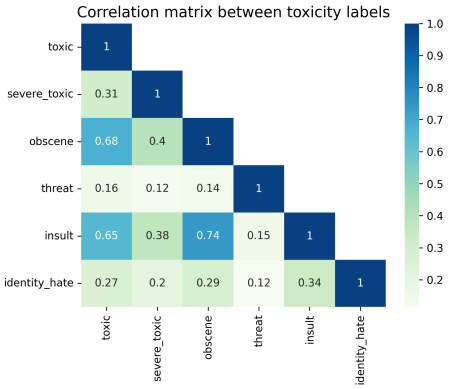


Figure 3.4: Correlation matrix over toxicity labels for the toxic-comments

It is important to note that the Jupyter notebook explored in [9] is currently outdated, as it references an earlier version of the dataset containing approximately half the number of comments compared to the updated dataset used in this work.
Regarding conformal prediction, the efficiency of the ground truth label sets (calculated across the entire test dataset) is approximately 1.114. When considering only the subset of toxic comments, the efficiency increases to 2.124. These values act can be used as a weak baseline for assessing if the framework is either under- or over-predicting labels.

3.2 Data Splitting

To assess the performance of both the classification model and the conformal prediction pipeline, the dataset was partitioned into two unique data splits using stratified splitting based on the toxic labels. Stratification is especially important given the multi-label nature of the dataset and the significant imbalance between toxic and non-toxic comments. This stratified splitting ensures that each subset maintains a representative distribution of label occurrences and co-occurrences.
The two data splitting strategies were designed to explore how different allocations of training and calibration data affect model and uncertainty estimation performance. The first split, referred to as the fine-tuning heavy split or data split I, allocates 60% of the data to fine-tuning, with 15% each for validation and calibration, and 10% for testing. This configuration is aims to maximize classification performance by providing the model with more labelled data to learn from, while still reserving enough data for reliable evaluation and calibration.
In contrast, the calibration heavy split shifts focus on conformal prediction by dedicating 60% of the data to the calibration set, with only 20% used for fine-tuning

and the remainder split between validation (10%) and testing (10%). This split investigates how a larger calibration set influences the quality and reliability of the conformal prediction sets, particularly in estimating nonconformity scores.
Both splits preserve the toxic label distribution through stratification, allowing for a fair comparison of how different data allocation strategies affect predictive performance and uncertainty quantification. Additionally, the test set was separated first in both configurations with a fixed seed, ensuring that the test set remains identical between splits. The structure and label distribution of these two splitting strategies are visualized in Figures 3.5 and 3.6.

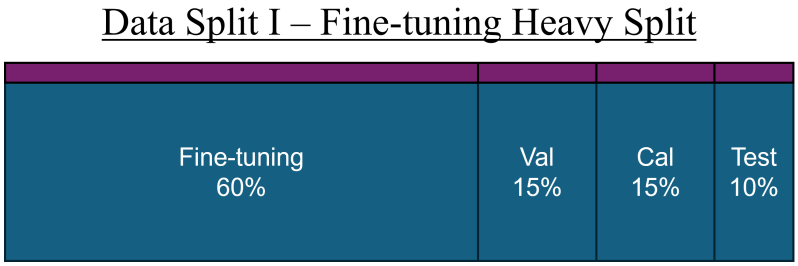


Figure 3.5: Dataset distribution for data split I prioritizing a larger fine-tuning set for increased model performance. With the colours purple and blue showcasing that the data is stratified with respect to toxicity of the comments.

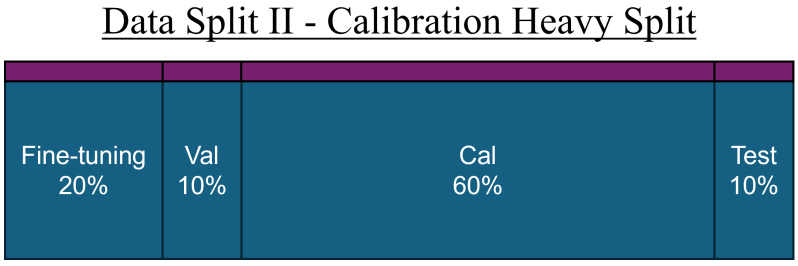


Figure 3.6: Dataset distribution for data split II prioritizing a large calibration set for conformal prediction. With the colours purple and blue showcasing that the data is stratified with respect to toxicity of the comments.

3.3 LLM Selection and Setup

3.3.1 Model Selection and Hyperparameter Configuration

Based on the comparative analysis of model performance metrics (Table A.3), the BERT model consistently achieved competitive performance across all epochs, with notably strong subset accuracy and precision scores. While RoBERTa and ELEC-

TRA also performed well, BERT's stable performance and extensive research support made it a robust choice for this study.
For fine-tuning, I used the `TrainingArguments` class from the Hugging Face Transformers library to configure training. The hyperparameters were selected to balance computational efficiency with model performance, ensuring consistency across all models. Specifically, the hyperparameters were set to:

- **Training epochs**: 3, based on convergence behaviour observed during evaluation and previous epoch testing.
- **Batch sizes**: 16 for both training and evaluation, optimizing GPU memory usage while maintaining reasonable training speed.
- **Learning rate**: 1×10^{-5}, a common value for fine-tuning and **Weight decay**: 0.01 to mitigate overfitting.

3.3.2 LLM Text Classifier Model Setup

Aside from the BERT model [5], both RoBERTa and ELECTRA models were trained using the same hyperparameters. However, the BERT model was chosen due to the large amount of prior research on it as well as its marginally better accuracy scores compared to the other models as seen in A.3. The model was implemented as described by the pseudocode in Algorithm 1.

Algorithm 1: Setup of the Multi-Label LLM Text Classification Model

Input: Multi-label text-based dataset

1. Split data: Training, validation, calibration, and test sets with toxicity stratification to account for label imbalances.
2. Compute label weights across the training dataset to allow training to account for label rarity.
3. Preprocess labels into multi-hot encoded vectors.
4. Train the LLM across several epochs:
   (a) Use a sigmoid activation function.
   (b) Train and validate loss using Binary Cross Entropy with label weights.
5. Validate and save models tensors across epochs.
6. Extract model tensors from the best-performing validation evaluation to avoid overfitting as seen in A.2b.

Output: Fine-tune LLM with tensors from best evaluation

3.4 Creation of Conformal Calibration Sets

In this section, I describe the process of creating the calibration sets required for conformal prediction using the multi-label classification framework outlined in this thesis. As discussed in Section 2.3.3, Inductive Conformal Prediction (ICP) relies

on splitting the available data into proper training and calibration subsets. This enables the construction of valid prediction sets by computing nonconformity scores on the calibration data and applying them to test examples.
For each label ℓ, the calibration set consists of a set of instances $Z_{cal} = \{(x_i, y_i^\ell)\}$, where x_i is the input text and y_i^ℓ is the ground-truth label indicator (0 or 1). The trained large language model is then used to compute the logits (pre-activation outputs) for each example in the calibration set. These logits are transformed into probabilities using the sigmoid activation function, yielding label-independent probabilities p_i^ℓ.
The nonconformity score α_i^ℓ for each instance is then computed as the inverse of the model's confidence in the correct label. Specifically, for positive labels ($y_i^\ell = 1$), $\alpha_i^\ell = 1 - p_i^\ell$ captures the model's lack of confidence in predicting the true label.
These nonconformity scores are then stored for each label ℓ and later used during the test phase to compute thresholds and construct the final conformal prediction sets. This process ensures that the calibration sets are properly aligned with the independent binary relevance approach adopted in this thesis (Subsection 2.3.4).

3.4.1 Empty and Full Prediction Sets

As a consequence of modifying the dataset by the addition of the non-toxic label, the true labels of comments are now somewhat constrained. Where the true labels of a comment will never be both toxic and non-toxic at the same time. Thus predicted comments containing both the non-toxic label and one of the six toxic labels can be considered uncertain cases. Empty prediction sets occur where none of the non-conformity scores fall below their respective thresholds, meaning that the conformalized classifier has rejected all possible labels. Conversely, full prediction sets is where every label is accepted, resulting in predictions that contain all toxic labels along with the non-toxic label. Both cases can occur and are dependent on the chosen level of significance: lower values correlate with overprediction (full sets), while higher values correlate with underprediction (empty sets).

3.5 Full Conformal Prediction LLM Text Classification Pipeline

Figure 3.7 illustrates the overall dataflow and conformal prediction framework used for multi-label text classification in this thesis. The process is structured into four sequential steps, each representing a key stage in building a prediction pipeline that combines both model performance and calibrated uncertainty.
First, a pre-trained BERT-base-uncased (LLM) is fine-tuned and validated on the labelled dataset to adapt it to the specific classification task as described previously

in subsection 3.3.2. In the second step, a calibration set is used to compute non-conformity scores based on the model's output probabilities (sigmoid transformed logits), which are used as a measure of how strange each predicted label is. Third, a significance level ϵ is selected to determine label-specific score thresholds with (E,δ) validity, effectively controlling the error rate of the prediction sets. Finally, the calibrated thresholds are then applied to test examples to produce prediction sets with label-wise coverage guarantees ensuring that the true labels are included with high probability. Through this process, the framework allows for interpretable, controllable multi-label predictions with quantified uncertainty

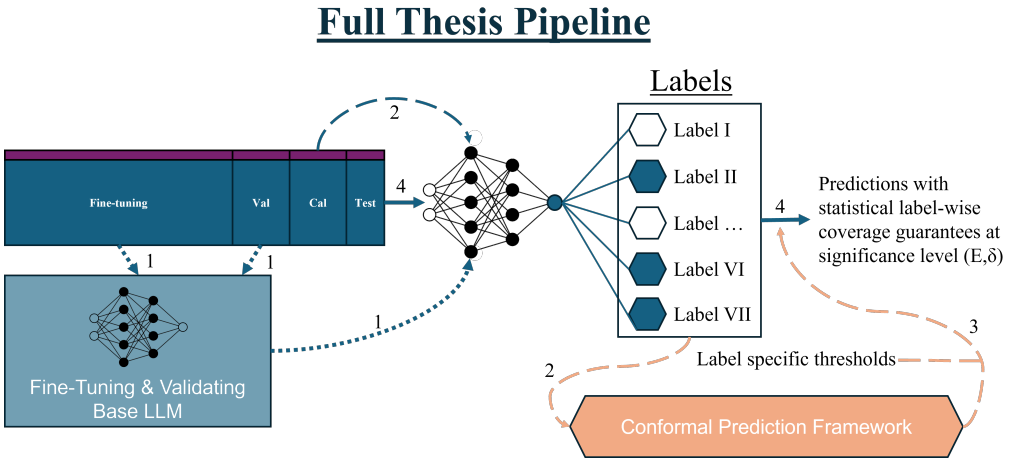


Figure 3.7: Flowchart showcasing the dataflow and conformal prediction framework for the multi-label text classification in this thesis. The process is divided into four steps consisting of (1) Fine-tuning and validating the LLM on the dataset, (2) computing a calibration set of non-conformity scores, (3) choosing a level of significance ϵ to determine label-specific thresholds and (4) producing predictions sets on the test data with label-wise coverage guarantees.

4

Results

This chapter presents the results of the tests conducted using the BERT-based large language model (LLM) for multi-label text classification of Wikipedia toxicity comments. The results include both model performance metrics and evaluations of the conformal prediction framework. The performance metrics—such as subset accuracy, precision, and recall are reported at the macro level and recall is additionally provided at per label to provide a in depth understanding of the model's capabilities. Additionally, the chapter compares the effects of different data splits (Datasplit I and II) on model training and calibration, offering insights into how data allocation impacts both base model performance and uncertainty quantification.

4.1 LLM Multi-label classification metrics

The performance of the BERT-based multi-label classifier was evaluated across two model configurations (Model 1 and Model 2) trained on the two different data splits presented in the method Subsection 3.2. Table 4.1 reports macro-level evaluation metrics, while 4.2 presents per-label accuracy to highlight the performance on individual toxicity classes. The models were trained on different epoch schedules due to disparity between the amount of training data in the splits, and were made to compare how much of available data can be used for calibration and fine-tuning respectively.

Table 4.1: BERT model 1 and model 2 macro metrics, all of f1, recall and precision are calculated with macro averaging.

Model	Model 1 (Epochs 1–3)			Trend	Model 2 (Epochs 5, 10, 15)			Trend
Epoch	1	2	3		5	10	15	
Log Eval Loss (ln)	-5.9935	-6.0106	-6.0064	↘↗	-5.8609	-5.5492	-5.5018	↗
Subset Accuracy	0.9191	0.9247	0.9223	↗↘	0.9206	0.9064	0.9137	↘↗
Precision (Macro)	0.6532	0.6548	0.7287	↗	0.7178	0.6774	0.7024	↘↗
Recall (Macro)	0.7579	0.7784	0.7247	↗↘	0.7102	0.7431	0.7271	↗↘

Table 4.2: BERT label-wise accuracy across the label set for Model 1 (Datasplit I) and Model 2 (Datasplit II). Model 1 is trained for 3 epochs, Model 2 for 15 epochs.

Model	Model 1 (Epochs 1–3)			Trend	Model 2 (Epochs 5, 10, 15)			Trend
Epoch	1	2	3		5	10	15	
Non-Toxic	0.9669	0.9687	0.9677	↗↘	0.9658	0.9551	0.9606	↘↗
Toxic	0.9652	0.9683	0.9670	↗↘	0.9648	0.9538	0.9594	↘↗
Severe Toxic	0.9904	0.9908	0.9901	↗↘	0.9912	0.9893	0.9901	↘↗
Insult	0.9717	0.9738	0.9746	↗	0.9748	0.9717	0.9736	↘↗
Obscene	0.9803	0.9818	0.9823	↗	0.9788	0.9788	0.9792	→
Threat	0.9978	0.9975	0.9974	↘	0.9974	0.9972	0.9973	↘↗
Identity Hate	0.9929	0.9934	0.9929	↗↘	0.9918	0.9920	0.9927	↗

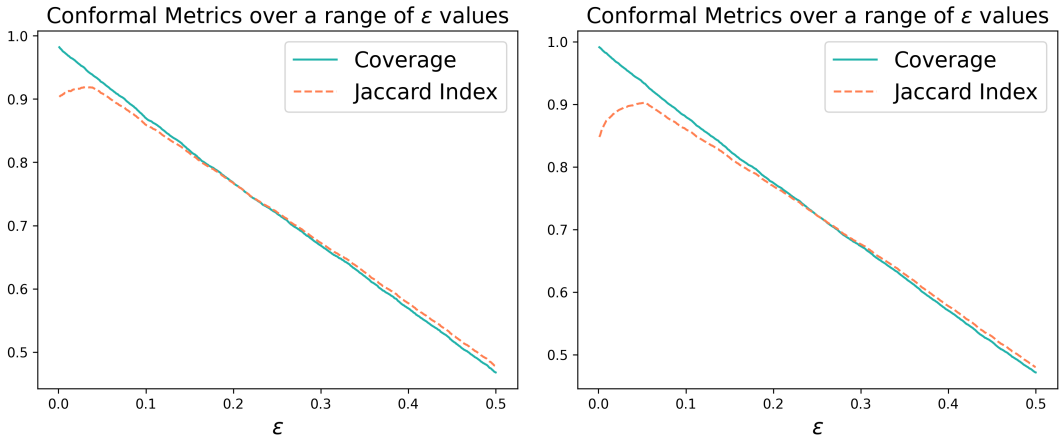
4.2 Conformal Prediction Evaluation

In this section, we have our significance level ϵ in the range 0.001 to 0.5 which is ensured validity via adjustment with Equation (2.7) based on the size of label-wise calibration sets with a predefined δ value of 0.05. Thereby creating a conformal predictor for each label valid with respective E values: E_1,...,E_ℓ. The effect of this shift for a few ϵ values is can be seen in Tables A.1 & A.2 with respect to both label and data split.

4.2.1 Conformal Classification Metrics across ϵ values

Figure 4.1a and 4.1b show that the coverage closely follows the identity line for test predictions in both models. This in turn provides empirical confirmation that the coverage guarantees from the conformal prediction are valid as expected although without rigorous mathematical proof.

For predictive quality, the Jaccard Index provides a useful measure of the overlap between the predicted and ground truth label sets. Both models display a single peak in the Jaccard Index as observed in the same figures, although at different significance levels, indicating the point of maximal alignment between the predicted and true labels. However, these peaks differ noticeably: model 1 reaches a higher maximum, the peak is narrow and appears early along the x-axis before quickly going towards the unit line. In contrast, the peak for model 2 is wider, with a more gradual climb, hitting its maximal values just above 0.9. Here it can be seen that model 1's Jaccard Index is at higher values in the range of significance levels between 0.001 and 0.1 whereafter they converge towards the same values.



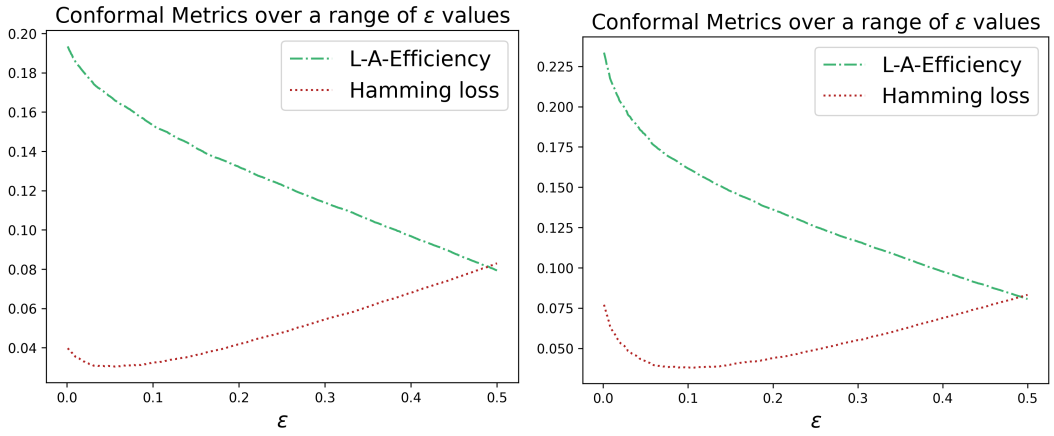
(a) Coverage & Jaccard Index for Model 1

(b) Coverage & Jaccard Index for Model 2

Figure 4.1: Coverage and Jaccard Index across significance levels from 0.001 to 0.5 with a step size of 0.001 and with each ϵ value adjusted with respect to label specific calibration set size and delta value.

Determining which of these metrics is most informative depends heavily on the dataset characteristics. In cases where the label-averaged efficiency is low, meaning relatively few ground truth labels are captured in the prediction sets, Hamming loss can be more useful, as it continues to account for correct rejections. Inversely, the Jaccard Index becomes less meaningful when label-averaged efficiency is close to 1, as almost all ground truth labels are included, and minor differences in false positives or calibration effects can instead dominate the metric.

Additionally, as the significance level increases towards 0.5, a divergence appears between these metrics, particularly for Model 2. This divergence reflects the different priorities each metric overlap vs. precision and showcases the importance of using these complementary metrics when evaluating multi-label predictive performance under conformal prediction.



(a) L-A-Efficiency & Hamming loss for model 1

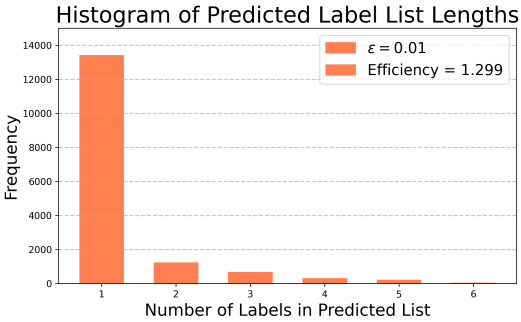
(b) L-A-Efficiency & Hamming loss for model 2

Figure 4.2: Label-Averaged Efficiency and Hamming loss across significance levels from 0.001 to 0.5 with a step size of 0.001 and with each ϵ value adjusted based on Equation 2.7 with respect to label specific calibration set size and a delta value of 0.05.

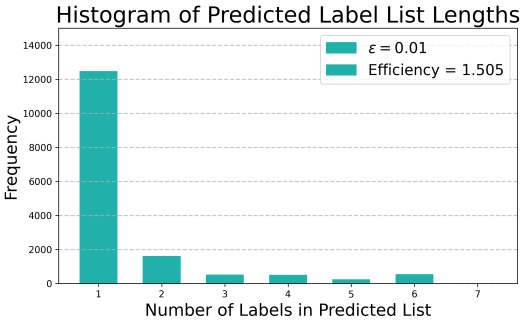
4.2.2 Conformal Prediction Set Efficiency

For a further exploration of the prediction set efficiency (the size of predicted label sets) of the two models Figure 4.3 presents the predicted label distributions across the test set. In all of the histograms the effect of the label imbalances previously shown in Figures 3.1 and 3.2 becomes apparent. This is seen by the clear majority of single label instances for all three significance levels for both models. On the other side there is full prediction sets appearing at significance level 0.01 for the calibration heavy model 2 showing an overestimation of label-predictions to ensure coverage in edge cases.

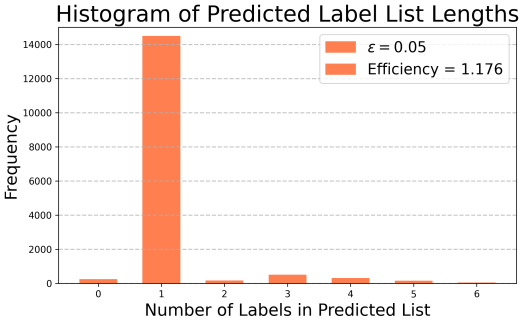
Empty prediction sets start appearing at significance level 0.05 for the fine-tuning heavy model 1 and for both models encounter a high volume of these at significance level 0.2. Between the models the largest difference in efficiency can be seen in the $\epsilon = 0.01$ case where model 1 has a 0.2 lower efficiency value than model 2. This suggests that model 2 likely overestimates the number of labels on out of distribution test cases, as it has been trained on a smaller subset of the dataset



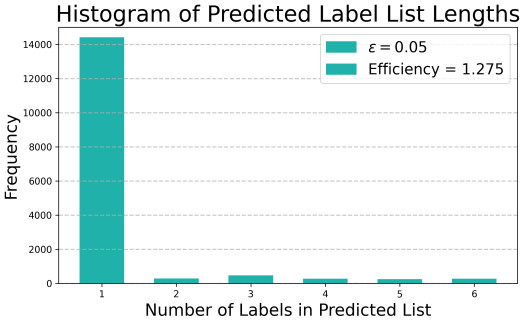
(a) Histogram of prediction set sizes $\epsilon = 0.01$ for model 1



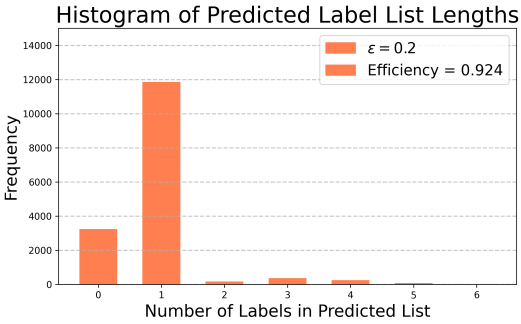
(b) Histogram of prediction set sizes $\epsilon = 0.01$ for model 2



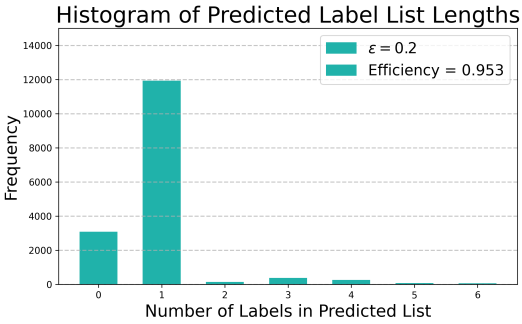
(c) Histogram of prediction set sizes $\epsilon = 0.05$ for model 1



(d) Histogram of prediction set sizes $\epsilon = 0.05$ for model 2



(e) Histogram of prediction set sizes for $\epsilon = 0.2$ for model 1



(f) Histogram of prediction set sizes for $\epsilon = 0.2$ for model 2

Figure 4.3: Histograms over predicted set size in the test set for ϵ values 0.01, 0.05 and 0.2 showing model 1 on the left side and model 2 on the right side.

5

Discussion

This chapter discusses the main findings of the thesis, placing them within the broader context of multi-label text classification, large language models (LLMs), and uncertainty estimation. By applying conformal prediction to LLM outputs in a toxicity classification task, the aim was to evaluate how reliably these models can produce interpretable prediction sets. The analysis considers both empirical results and theoretical implications, highlighting trade-offs between coverage and prediction set size. Key limitations and assumptions are addressed, along with reflections on the practical relevance of the approach.

5.1 Calibration Across Time and Dataset Drift

The reliability of conformal predictors depends not only on accurate calibration at a specific point in time but also on the stability of the underlying data distribution. In natural language tasks, changes in language use, social norms, or the visibility of certain topics often cause the test data to differ from the training and calibration data. This type of temporal change is commonly referred to as dataset drift.

For example, a phrase that was once seen as neutral may later gain a toxic connotation. Similarly, identity-based slurs that were previously uncommon might become more frequent due to changes in political or cultural discourse. These shifts can weaken the usefulness of nonconformity scores calculated from outdated calibration data, leading to a breakdown in the formal coverage guarantees of conformal prediction.

To manage this, future systems might benefit from periodic recalibration using more recent labelled data, or from adaptive methods such as sliding-window calibration or incorporating timestamp information into the model input. While these strategies could help maintain calibration quality in changing environments, they also add implementation and computational complexity. In many cases, they may require a relaxation of the strict distribution-free assumptions of standard inductive conformal prediction.

5.1.1 Exchangeability and Label Drift

Conformal prediction also relies on the assumption of exchangeability between calibration and test data, as discussed in Subsection 2.3.3. This means that the two sets should come from the same distribution, and that the order of data points should not matter.
In this thesis, the assumption may be challenged by inconsistencies in the manual annotation of the dataset. Annotators might have applied different thresholds for what qualifies as toxic, and these thresholds could shift over time or vary between annotated batches. Moreover, since the dataset contains isolated comments without conversational context, the LLMs may misinterpret sarcastic or indirect language due to lack previous conversational context.
These issues contribute to what is often referred to as label drift: a gradual change in how labels are applied over time. As a result, the calibration set may no longer reflect the characteristics of the current test data. This highlights the importance of consistent labelling practices, as well as the potential benefits of incorporating human review for uncertain or ambiguous predictions.

5.2 Impact of Dataset Imbalance on Coverage Validity

One of the key challenges in applying conformal prediction to multi-label text classification is the impact of label imbalance. In the dataset used in this thesis, the vast majority of comments are non-toxic, resulting in a roughly 9:1 ratio between non-toxic and toxic instances. This imbalance has significant implications for the estimation of nonconformity scores and for the calibration of prediction sets. Since conformal prediction relies on label-wise empirical distributions of these scores, labels with limited representation in the calibration set contribute to poorly estimated thresholds, which can lead to suboptimal or unstable prediction sets for those labels. Additionally, imbalanced label frequencies affect not only the statistical validity but also the interpretability of the produced conformal sets. In particular, for labels appearing with high frequency (e.g., *toxic* or *obscene*), the thresholds are computed over a richer distribution, leading to a more fine-grained control over coverage and efficiency. In contrast, rare labels may receive overly cautious or excessively permissive thresholds due to insufficient calibration samples. This phenomenon can manifest as inflated prediction sets for rare labels in order to maintain the theoretical guarantees, reducing practical utility. As a result, even though coverage validity may be mathematically preserved after adjusting the significance level using Equation (2.7), its real-world applicability must be interpreted with care especially in industrial or safety critical contexts.

5.2.1 Handling Rare Labels and Minimum Significance

As shown in Table A.1, rare labels present a clear challenge for conformal calibration. The *threat* label, for example, appears only 62 times in the calibration set for data split I. This limited sample size results in a minimum valid significance level of around 15.5% in order to maintain (E, δ)-validity. In applications that require high-confidence predictions, such as content moderation or legal review, this level of tolerated error may be very difficult to justify. Even so, the ability of conformal prediction to abstain from making uncertain predictions offers a practical advantage. Instances where no labels pass the threshold can be flagged for manual inspection, helping maintain trust in the system's reliability.

In many industrial contexts, automation efforts are primarily focused on high-volume, high-certainty cases, with low-confidence or rare-label instances deferred to human reviewers. This strategy mirrors the approach taken in [22], where labels with fewer than 1000 examples were excluded from automatic processing to simplify the workflow. The results in this thesis support a similar approach. Through the binary relevance framework, each label's predictor can independently abstain when calibration is weak or insufficient—enabling graceful degradation rather than failure. This division of labour between statistical automation and human oversight suggests a robust path forward for deploying MLCP systems in noisy or skewed real-world cases.

5.3 Significance Level Optimization Trade-offs in Conformal Prediction

A central hyperparameter in conformal prediction is the significance level ϵ, which governs the trade-off between prediction set conservativeness and coverage. Smaller values of ϵ yield larger prediction sets with higher coverage guarantees, ensuring that true labels are included with high probability. However, this often comes at the cost of efficiency, interpretability, and practical usefulness—especially in settings where overly broad predictions introduce ambiguity or hinder downstream decisions.

Conversely, increasing ϵ tightens the prediction sets, improving their precision and reducing average set size, but at the risk of omitting true labels and thereby violating coverage expectations. This trade-off is clearly visualized in Figure 4.1a for data split I, where the Jaccard Index shows a distinct peak around $\epsilon = 0.05$. This value appears to represent an optimal balance point for the dataset used in this thesis, combining sufficient label inclusion with minimal overprediction. The ability to tune ϵ flexibly allows practitioners to adapt conformal prediction systems to different risk tolerances, user expectations, or regulatory demands, making it an essential control parameter in any deployment setting.

5.3.1 Empty Predictions and Set Size Behaviour

As shown in Figure 4.3, the structure of prediction sets changes significantly with the choice of ϵ. For lower values such as $\epsilon = 0.01$ or 0.02, every prediction set includes at least one label, often leading to overcoverage or redundant label inclusion. These low thresholds ensure that rare or borderline cases are not missed, but result in larger sets that may decrease interpretability and introduce uncertainty into classification decisions.

Only when $\epsilon \geq 0.05$ do empty prediction vectors begin to appear—instances where the model abstains from making any confident prediction, producing a set such as $[0, 0, \ldots, 0]$. This behaviour demonstrates one of the practical strengths of conformal prediction: its ability to reject uncertain inputs gracefully rather than forcing unreliable predictions. The appearance of empty sets corresponds closely with the observed peak in Jaccard Index around $\epsilon = 0.05$, as shown in Figure 4.1a. This suggests that $\epsilon = 0.05$ not only provides a good statistical trade-off but also marks a functional transition point where the model begins to express calibrated uncertainty through abstention.

5.3.2 Comparison between data splits

The two data-splitting strategies employed in this thesis, data split I (fine-tuning heavy) and data split II (calibration heavy), were designed together to investigated the trade-off between model accuracy and uncertainty calibration. Data split I allocated more data to fine-tuning the base LLM, resulting in a better-trained classifier with stronger macro-level performance across most epochs, as shown in Table 4.1. This configuration produced higher Jaccard Index values at low significance levels and showed more consistent behaviour across different values of ϵ. In contrast, data split II used a larger calibration set, which improved the stability of nonconformity thresholds but led to reduced overall performance due to a smaller training set.

The difference in efficiency and coverage behaviour across the two splits is most apparent at the extremes of the significance levels shown. As shown in Figure 4.3, model 2 (data split II) tends to produce larger prediction sets at low ϵ values, suggesting overestimation likely due to a less-confident underlying classifier. Conversely, model 1 (data split I) achieves tighter prediction sets with fewer full-label predictions, especially at $\epsilon = 0.01$, while still maintaining valid coverage. These findings suggest that while a larger calibration set can enhance theoretical reliability, a well-trained base model remains crucial for practical effectiveness, especially in settings where prediction efficiency and semantic precision are as important as formal validity.

5.3.3 Jaccard Index vs Hamming Loss

Jaccard Index and Hamming Loss offer complementary views of performance in multi-label classification. The Jaccard Index measures the overlap between predicted and true label sets, making it useful when overall set similarity is important, such as in tagging or moderation tasks where overprediction is costly. As seen in Figures 4.1a and 4.1b, it peaks at intermediate significance levels, suggesting an optimal balance between precision and recall.

Hamming Loss, by contrast, evaluates performance at the label level, calculating the average number of incorrect label predictions per instance. It is less sensitive to overall set composition but provides a clearer picture of individual label accuracy. Figures 4.2a and 4.2b show that Hamming Loss increases more gradually with ϵ, providing stable feedback even as prediction confidence varies.

These differences are especially relevant when scaling to larger label spaces. While both metrics align well in the 7-label setting used in this thesis, Jaccard Index becomes less reliable in high-dimensional cases, where even small mismatches inflate the set union. Hamming Loss, on the other hand, remains interpretable and consistent as the number of labels grows, making it better suited for domains like topic classification or medical coding. For large-scale applications, Hamming Loss and related label-wise metrics may offer more practical guidance than set-based measures like the Jaccard Index.

5.4 Conclusions

This thesis explored the application of conformal prediction methods to large language models (LLMs) for multi-label text classification tasks, with a focus on classifying toxic comments in a Wikipedia dataset. The goal was to enhance the trustworthiness of model predictions by quantifying uncertainty in the form of prediction sets.

The findings indicate that conformal prediction can effectively calibrate LLM outputs, producing prediction sets that contain the true labels with high probability while maintaining reasonable efficiency. The results demonstrate that conformal methods, particularly in multi-label settings, offer a principled approach to uncertainty estimation even when applied to powerful pre-trained models. We also observed trade-offs between the size of prediction sets and coverage guarantees, which emphasizing the need for task-specific calibration.

Overall, this thesis shows that conformal prediction can be a valuable component in enhancing the transparency and reliability of LLM-based classifiers, especially in sensitive domains such as content moderation.

5.5 Future Work

There are several interesting directions for future work that could build on the results and ideas presented in this thesis:

- **Testing on Other Datasets and Domains:** This thesis focused on Wikipedia toxicity comments, but applying the same approach to other datasets could be of interest. Examples of this would be tagging emails, legal documents, or medical reports, which could show how well the method generalises. It would also be interesting to see how it performs in non-NLP multi-label settings, such as image tagging or medical prediction based on blood tests.
- **Comparing Different MLCP Frameworks:** While this thesis used the binary relevance approach, there are other ways to handle multi-label conformal prediction, such as Label Powerset (LCP) and Instance Reproduction (IR). A more thorough comparison of these methods, especially for text classification tasks, could highlight strengths and weaknesses depending on the use case.
- **Trying Alternative Nonconformity Measures:** The current setup used sigmoid scores as the basis for calibration. Future work could compare it against other nonconformity scores such as sample frequency, normalized entropy, or semantic similarity treated in [19], which might give better results for edge cases.
- **Human-in-the-Loop Implementation:** One of the strengths of conformal prediction is that it can abstain when uncertain. This makes it a good fit for systems where humans can step in to review flagged cases. Exploring how such a system could be implemented and used would offer practical insights into how these systems are applied and trusted in the real world.

Bibliography

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd. 2025, Online manuscript released January 12, 2025. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. DOI: 10.48550/ARXIV.1301.3781.

[3] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014. DOI: 10.3115/v1/d14-1162.

[4] A. Vaswani *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805.

[6] Y. Liu *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. DOI: 10.48550/ARXIV.1907.11692.

[7] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, 2020. DOI: 10.48550/ARXIV.2003.10555.

[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[9] D. Carlos, *Exploring the toxicity of wikipedia comments*, Kaggle Notebook, Accessed: 2025-01-16, 2018. [Online]. Available: https://www.kaggle.com/code/djcarlos/exploring-the-toxicity-of-wikipedia-comments.

[10] H. Face, *Transformers v3.0.2 - automodelforsequenceclassification*, https://huggingface.co/transformers/v3.0.2/model_doc/auto.html, Accessed: 2025-04-04, 2020.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, ch. 5, http://www.deeplearningbook.org.

[12] Y. Chen, L. Li, W. Li, Q. Guo, Z. Du, and Z. Xu, "Fundamentals of neural networks," in *AI Computing Systems*. Elsevier, 2024, pp. 17–51, ISBN: 9780323953993. DOI: 10.1016/b978-0-32-395399-3.00008-1.

[13] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, Oct. 2020. DOI: 10.30534/ijatcse/2020/175942020.

[14] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," 2016. DOI: 10.48550/ARXIV.1609.00288.

[15] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, Mar. 2021. DOI: 10.1007/s10994-021-05946-3.

[16] L. Huang *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023. DOI: 10.48550/ARXIV.2311.05232.

[17] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World, Second Edition*, English (US). Springer International Publishing, Jan. 2022, Publisher Copyright: © Springer Verlag New York, Inc. 2005., ISBN: 9783031066481. DOI: 10.1007/978-3-031-06649-8.

[18] D. J. Aldous, I. A. Ibragimov, and J. Jacod, *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983*. Springer, 2006, vol. 1117.

[19] J. Su, J. Luo, H. Wang, and L. Cheng, *Api is enough: Conformal prediction for large language models without logit-access*, 2024. arXiv: 2403.01216 [cs.CL].

[20] L. Maltoudoglou, A. Paisios, L. Lenc, J. Martínek, P. Král, and H. Papadopoulos, "Well-calibrated confidence measures for multi-label text classification with a large number of labels," *Pattern Recognition*, vol. 122, p. 108 271, 2022.

[21] H. Wang, X. Liu, I. Nouretdinov, and Z. Luo, "A comparison of three implementations of multi-label conformal prediction," in *Statistical Learning and Data Sciences*. Springer International Publishing, 2015, pp. 241–250, ISBN: 9783319170916. DOI: 10.1007/978-3-319-17091-6_19.

[22] A. Borg, M. Boldt, and J. Svensson, "Using conformal prediction for multi-label document classification in e-mail support systems," in *Advances and Trends in Artificial Intelligence. From Theory to Practice*. Springer International Publishing, 2019, pp. 308–322, ISBN: 9783030229993. DOI: 10.1007/978-3-030-22999-3_28.

A

Appendix 1

A.1 Jaccard Index with Labels Sets

Looking back at the Jaccard index defined in Subsection 2.3.6, with the one-hot encoded labels Jaccard index with labels being: non toxic, toxic, severe toxic, insult, obscene, threat, and identity hate. In this case of label sets this can be visualized in Figure A.1. An example calculation of Jaccard with the true label vector set being [0, 0, 1, 1, 0, 0, 1] and the predicted vector set after applying conformal prediction being [0, 0, 1, 1, 1, 1, 0] would end up as:

$$A \cap B = \frac{[0,0,1,1,0,0,1]}{[0,0,1,1,1,1,0]} = 2, \quad A \cup B = [0,0,1,1,1,1,1] = 5, \quad J_{acc} = \frac{2}{5} = 0.4$$

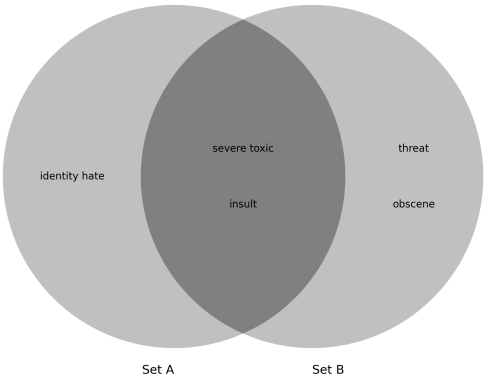


Figure A.1: Venn diagram over two possible vector sets in the data.

A.2 Significance Level Shifts for Labels Accounting for Calibration Set Size

Table A.1: Table of E values for adjusted validity across the label-wise calibration sets with a fixed δ value 0.05. The adjustment term Δ seen next to ϵ in the RHS of equation 2.7 δ. CSS in the table is the calibration set size for each respective label and conformal predictor.

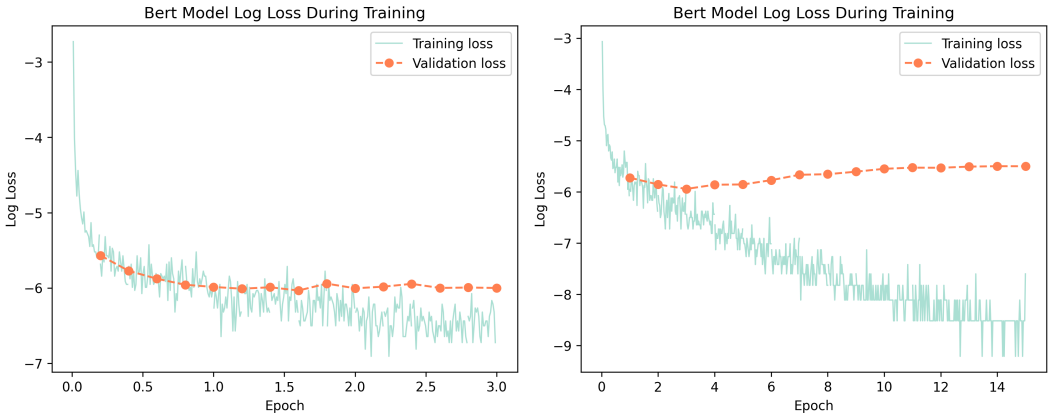
label	non toxic	toxic	severe toxic	insult	obscene	threat	identity hate
CSS	21515	2294	259	1200	1263	62	209
Δ	0.00834	0.02555	0.07605	0.03533	0.03444	0.15543	0.08466
ϵ	E_1	E_2	E_3	E_4	E_5	E_6	E_7
0.01	0.01834	0.03555	0.08605	0.04533	0.04444	0.16543	0.09466
0.02	0.02834	0.04555	0.09605	0.05533	0.05444	0.17543	0.10466
0.05	0.05834	0.07555	0.12605	0.08533	0.08444	0.20543	0.13466
0.1	0.10834	0.12555	0.17605	0.13533	0.13444	0.25543	0.18466
0.2	0.20834	0.22555	0.27605	0.23533	0.23444	0.35543	0.28466
0.5	0.50834	0.52555	0.57605	0.53533	0.53444	0.65543	0.58466

Table A.2: Table of E values for adjusted validity across the label-wise calibration sets with a fixed δ value 0.05. The adjustment term Δ seen next to ϵ in the RHS of equation 2.7 δ. CSS in the table is the calibration set size for each respective label and conformal predictor.

label	non toxic	toxic	severe toxic	insult	obscene	threat	identity hate
CSS	86052	9181	988	4740	5140	289	830
Δ	0.00834	0.02555	0.07605	0.03533	0.03444	0.15543	0.08466
ϵ	E_1	E_2	E_3	E_4	E_5	E_6	E_7
0.01	0.01834	0.03555	0.08605	0.04533	0.04444	0.16543	0.09466
0.02	0.02834	0.04555	0.09605	0.05533	0.05444	0.17543	0.10466
0.05	0.05834	0.07555	0.12605	0.08533	0.08444	0.20543	0.13466
0.1	0.10834	0.12555	0.17605	0.13533	0.13444	0.25543	0.18466
0.2	0.20834	0.22555	0.27605	0.23533	0.23444	0.35543	0.28466
0.5	0.50834	0.52555	0.57605	0.53533	0.53444	0.65543	0.58466

A.3 Multi-label Classification Performance for Considered LLMs

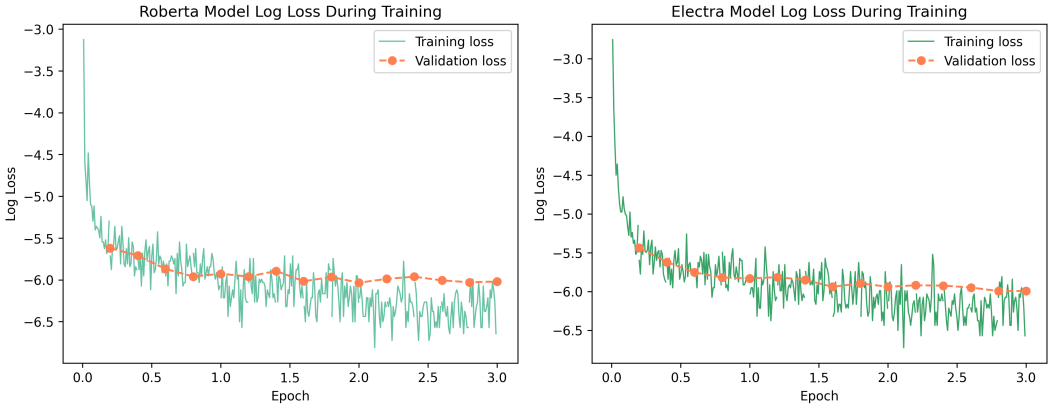
Before creating the full conformal prediction pipeline for the multi-label case three different models were evaluated for their predictive performance. The models chosen were Bert, Deberta & Electra. Training of all of these models can be seen in Figures A.2a, A.2b, A.3a & A.3b. Additionally Bert due to its performance were also trained on the calibration heavy data split showcased in Figure A.2b.



(a) Training and validation loss for 3 epochs on data split I

(b) Training and validation loss for 15 epochs on data split II

Figure A.2: Training and validation loss for Bert model on both data splits for respective training and validation sets



(a) Training and validation loss for 3 epochs on data split I

(b) Training and validation loss for 3 epochs on data split II

Figure A.3: Training and validation loss for Roberta and Electra models on dataplit 1 for training and validation sets

Table A.3: Model performance metrics and validation loss across three epochs during fine-tuning with regards to data split I - fine-tuning heavy

Model	Epoch	Validation Loss	Subset Accuracy	Precision (Macro)	Recall (Macro)
Bert	1	0.30372	0.919118	0.757860	0.653184
Bert	2	0.46620	0.924716	0.778423	0.654780
Bert	3	0.45485	0.922335	0.724724	0.728683
Roberta	1	0.30193	0.920037	0.736345	0.679525
Roberta	2	0.36652	0.924256	0.755798	0.683694
Roberta	3	0.31043	0.920872	0.715743	0.752577
Electra	1	0.25818	0.920037	0.736345	0.679525
Electra	2	0.41728	0.924256	0.755798	0.683694
Electra	3	0.34969	0.920872	0.715743	0.752577

Table A.4: Model performance metrics and validation loss for BERT across 15 epochs with regards to data split II - calibration heavy

Epoch	Validation Loss	Subset Accuracy	Precision (Macro)	Recall (Macro)
1	0.00326	0.91567	0.73270	0.59413
2	0.00286	0.91875	0.70060	0.70225
3	0.00262	0.91749	0.72826	0.69245
4	0.00285	0.91278	0.68578	0.75269
5	0.00286	0.92063	0.71782	0.71015
6	0.00311	0.91988	0.72799	0.69909
7	0.00346	0.91153	0.67139	0.76781
8	0.00350	0.91812	0.71294	0.72612
9	0.00367	0.90870	0.69644	0.71290
10	0.00389	0.90644	0.67735	0.74309
11	0.00397	0.91385	0.69061	0.75010
12	0.00396	0.90990	0.68976	0.73901
13	0.00405	0.91580	0.69955	0.73231
14	0.00409	0.91404	0.70256	0.73018
15	0.00408	0.91366	0.70240	0.72714

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden
www.chalmers.se

CHALMERS