Active Learning for Cases of Low Data or High Number of Classes

Student: Ruben Mathew (rubenom2)

Introduction

One major task performed by Machine Learning models is Text Classification. Being able to take an arbitrary string of text and sort it into different categories is useful for many applications. However, training these algorithms may become difficult depending on the amount of data given to train, and the number of classes it has to sort between. This could potentially be solved through using Active Learning.

Active Learning (for text classification) is when a model will do its best to classify each string presented to it, but will take the top cases that it is unsure about and present it to a human. The human will then manually label these cases and the algorithm will use that information to better itself. This should help increase the accuracy of the model since it should clear up any confusion it has on similar tricky cases going forward. The model decides which cases to ask about using a query function.

The goal of this project is to test different query functions and to compare them against each other in multiple scenarios based on training data size and classes.

Inspiration

The inspiration for this project came from this paper titled “Active learning for reducing labeling effort in text classification tasks”. This paper seeks to test different algorithms for active learning and compares it to random sampling. I want to take this further and test it with varying amounts of classes and training data amounts to see if Active Learning performs better.

Dataset

The dataset I will be using is Amazon Review 2023. This dataset has about ~570 million reviews labeled by categories including ratings and item categories. This is the perfect dataset for this experiment because it has a great amount of data to test with, and has a multitude of different classification problems. We can test how Active Learning compares in a situation with a low number of classes (positive/negative review) or a high number of classes (each of the 30+ item type categories).

Model/Baseline

Unlike the paper, I will be using DistilBERT as the base model. I will use traditional fine-tuning methods for a baseline to compare against AL.

Evaluation

There are two main behaviours that the different algorithms will be evaluated on. One is accuracy and the other is training speed. Training speed is self-explanatory. Accuracy will depend on the classification done. Here are non-specific example tables that I would use for comparision:

Binary Classification (Positive/Negative Reviews)

Method Type	**Accuracy**	**Precision**	**F1 Score**	**Training Speed**
Method 1 - X amount of Data				
Method 2 - X amount of Data				
Method 3 - X amount of Data				
Method 1 - Y amount of Data				
Method 2 - Y amount of Data				
Method 3 - Y amount of Data				

Multi-Classification (Different Item Types)

Method Type	**Accuracy**	**Top-k accuracy**	**Weighted F1**	**Speed**
Method 1 - X amount of Data				
Method 2 - X amount of Data				
Method 3 - X amount of Data				
Method 1 - Y amount of Data				
Method 2 - Y amount of Data				
Method 3 - Y amount of Data				

Where each metric is calculated like so:

Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$	**Precision:** $\frac{TP}{TP+FP}$	**F1:** $2 \cdot \frac{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}}$
	Top K: $\frac{1}{N} * \sum_{i=1}^{N} y_i \in \mathbf{TopK}(p_i)$	**Weighted F1:** $\sum_{c=1}^{C} \frac{n_c}{N} * \mathbf{F1}(c)$

Where:
TP - True Positive, TN - True Negative, FP - False Positive, FN - False Negative
N - Total number of data
y_i - True classification (for multi-classification)
p_i - Predicted classification (for multi-classification)
TopK - Function that returns the Top K most predicted classes in p_i.
n_c - Number of data entries in class c
F1 - Function that returns the F1 score of a specific class.

Concluding Thoughts/Summary

This project will look into the performance of different Active Learning query functions/algorithms and compare them to each other and traditional fine tuning in various situations. The goal is to see if Active Learning can increase the performance of a model without substantially increasing training time, and in the absence of additional data. The hope is to see that with additional human guidance, the model would be able to converge on accurate answers for unclear cases.

I will not be downloading any code from any open source project. I will be using DistilBERT as a base model. This will be tested with the Amazon Review 2023 Dataset.

Related Works

Pieter Floris Jacobs, Gideon Maillette de Buy Wenniger, Marco Wiering, Lambert Schomaker: “Active learning for reducing labeling effort in text classification tasks”, 2021;