Prediction Stability of Text Classification Methods

Comparing Traditional and Deep Learning Methods

Orin Brown
University of Colorado Boulder
obbrown1@gmail.com

ABSTRACT

Text classification is a fundamental component of Natural Language Processing (NLP), which has a broad range of applications, including document search, sentiment analysis, chatbots and virtual assistants, machine translation, and many others. Text classification methods have evolved significantly over the past decades, going from rule-based approaches to the sophisticated machine learning models that are the current state of the art. This paper aims to investigate a key characteristic of these models: prediction stability.

INTRODUCTION

This paper aims to investigate the prediction stability of different text classification models. Prediction stability refers to the consistency and reliability of the classification results obtained from text classifiers when applied to different samples or variations in the input data. It aims to investigate the robustness of text classifiers and examine how consistent their predictions are across different instances of the same problem.

Prediction stability is crucial in text classification as it directly impacts the reliability and trustworthiness of the classification results. In real-world scenarios, it is essential to have text classifiers that produce consistent predictions that are robust in the face of small or insignificant variations in the input data or different samples. By evaluating and understanding the prediction stability of text classifiers, we can gain insights into their reliability and make informed decisions regarding their deployment in various applications, such as sentiment analysis, spam detection, and document categorization.

Existing text classification solutions often focus on maximizing accuracy without explicitly addressing prediction stability. While these solutions may perform well in terms of accuracy on specific datasets, they may lack consistency when applied to different samples or variations in the input data. Additionally, some classifiers may be sensitive to small changes in the input, leading to different predictions even for similar instances. These limitations pose challenges in deploying text classifiers in real-world applications where consistent and stable predictions are necessary.

RELATED WORK

Text classification methods have evolved significantly over the years, driven by advancements in machine learning and natural language processing techniques. “From the 1960s until the 2010s, traditional text classification models dominated. Traditional methods mean statistics-based models, such as Naïve Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM).”[1] These models suffer from a number of limitations, however. Firstly, they require hands-on work such as feature engineering, which is time consuming and prone to bias. Secondly, because they are purely statistical models, they fail to account for syntactic and semantic relationships between words or word tokens, which renders them unable to capture a great deal of the meaning communicated through language. Due to these limitations, current research is largely focused on deep learning methods which, while they have drawbacks of their own, address many of the issues that plague traditional approaches.

PROPOSED WORK

Datasets: This project will focus on datasets containing categorized news articles.

The project will utilize the r8 and BBC news datasets sourced from Kaggle. These datasets offer a diverse range of text samples across multiple categories, enabling a comprehensive analysis of prediction stability.

Tools: The project will employ Python as the primary programming language, along with relevant libraries for text analysis and machine learning tasks. A Jupyter notebook will be used as the development environment to facilitate code experimentation and documentation. The project's code and resources will be managed using GitHub for version control and collaboration.

Tasks: The primary task will be analyzing the prediction stability of traditional and deep learning classification methods. To this end, models of each category will be implemented and trained to a reasonable degree of accuracy. Once done, experiments will be run to determine how the gradual addition of new words affects model predictions.

Data Preprocessing and Warehousing: The project will involve preprocessing the text data to transform it into a suitable format for classification. Common preprocessing techniques like tokenization, stop word removal, stemming or lemmatization, and feature extraction will be employed. Additionally, the preprocessed data will be stored locally and managed using GitHub for efficient data warehousing and version control.

Statistical Analysis: The project will conduct a comprehensive statistical analysis of the text data. This will include examining word frequencies, average document length, unique vocabulary,

and other relevant statistical measures. The aim is to gain insights into the characteristics and distribution of the text data.

Visualization: Visualizations, such as charts and graphs, will be employed to display any patterns observed during the analysis. These visualizations will provide a clear understanding of the text data's characteristics and assist in identifying potential trends or patterns.

EVALUATION

The models will be evaluated across the dimensions of classification accuracy, training speed, and prediction stability.

Effectiveness: The primary effectiveness metric will be classification accuracy. It measures the percentage of correctly classified instances out of the total. Accuracy provides an overall measure of the classifier's ability to assign the correct class label to text samples.

Efficiency: Time/Resource Efficiency: This metric evaluates the computational efficiency of the text classifiers. It involves measuring the time taken to train the models and make predictions on new text samples. Additionally, resource consumption, such as memory usage, will be considered to ensure efficient utilization of computational resources.

Training Data Efficiency: This metric addresses the efficiency of the classifiers when limited training data is available. The project will explore the performance of the classifiers when training data is scarce or absent for specific categories. This will assess their ability to generalize and make predictions on unseen classes. This is an important metric that can assess a model's ability to accurately predict data from new domains with limited labeled data available, or from highly specialized domains with an abundance of jargon.

Experimental setup: The project will set up experiments to observe the behavior of prediction confidence when additional words are added to text samples classified with high confidence. The aim is to analyze how the classifiers' confidence levels change with the introduction of new words or variations in the input. By examining the prediction confidence behavior, insights can be gained into the stability and robustness of the classifiers.

The experimental setup will involve selecting a set of text samples that are classified with high confidence by the classifiers. Additional words will be added to these samples, and the classifiers' prediction confidence scores will be recorded. The behavior of the confidence scores, such as their variation or consistency, will be analyzed to understand how the classifiers handle modifications in the input and their level of confidence in the revised samples.

By incorporating these evaluation metrics and experimental setups, the project aims to provide a comprehensive assessment of the classifiers' effectiveness, efficiency, and prediction stability, contributing to a better understanding of their performance in real-world scenarios.

DISCUSSION

Timeline: There are roughly six weeks left in the term at time of writing, so I propose to divide the tasks as follows

- Data gathering – 1 week

  This has proved to be relatively accurate. I feel that I have enough data to train the models on. I may have come back to this step later if that proves not to be the case as I begin modeling.

- Research on models – 1 week

  This has run over its original time allotment, which I suppose should not be surprising. There are a seemingly endless number of different model architectures, and more tweaks and permutations on those.

- EDA and data cleaning – 1 week

  The data I have found is clean for the most part, and small enough in size that warehousing is not an issue. EDA is ongoing.

- Modeling and model tuning – 2 weeks

  I anticipate that this will run over its allotted time.

- Analysis and report writing – 1 week

  I plan to write and do preliminary analysis as I am building the models, so hopefully I can reclaim some time here.

Potential Challenges: Identifying an appropriate dataset could be a challenge. Identifying appropriate models that can produce classification estimates or probabilities may also take some time, but I am confident that there are at least two available. Sticking to the proposed timeline may also be a challenge.

Alternatives: If the proposed analysis proves to be impractical, I can shift my focus to some other metric in the text classification space. Perhaps investigating how well models can transfer learning from one corpus to another. Are the models sufficiently general to be able to do this well, or do they need training on each specific dataset?

CONCLUSION

TBD

REFERENCES

On the Sensitivity and Stability of Model Interpretations in NLP
https://aclanthology.org/2022.acl-long.188.pdf

A Survey on Text Classification Algorithms: From Text to Predictions
https://www.mdpi.com/2078-2489/13/2/83

Text Classification Algorithms: A Survey
https://medium.com/text-classification-algorithms/text-classification-algorithms-a-survey-a215b7ab7e2d

Li et al., A Survey on Text Classification: From Traditional to Deep Learning ACM Transactions on Intelligent Systems and TechnologyVolume 13Issue 2Article No.: 31pp 1–41https://doi.org/10.1145/3495162

[1] Li et al., A Survey on Text Classification: From Traditional to Deep Learning
ACM Transactions on Intelligent Systems and TechnologyVolume 13Issue 2Article No.: 31pp 1–41https://doi.org/10.1145/3495162