NLP Sentiment Analysis using Transformers

Project Report

Project NLP: DLBAIPNLP01

BSc. Applied Artificial Intelligence

August 5, 2023

Prepared By: Brandon Scott Copeland

Matriculation #: 32111637

Tutor: Anna Androvitsanea

Table of Contents

Introduction

From its inception during the 1950s until the present, Natural Language Processing (NLP) has both fascinated its users and researchers while ushering in a tidal wave of artificial intelligence development. Ironically, during the development of NLP, there was a point of stagnation where researchers felt that it wasn't worth it to continue working with such technologies. This led to one of the first AI winters where funding, research, and development of NLP ceased, particularly in the area of machine translation. However, during the 1980s, there was renewed interest in the NLP domain due to many factors. Increased compute power, larger storage, more powerful CPU options were just some of the reasons that lead to a reawakening of NLP as a bonafide discipline worthy of continued interest. Fast forward to the 2020s and NLP has exploded in both its implementation as well as its usage in common everyday applications. Whether it is recommender systems, sentiment analysis, or generative AI, NLP lies at the backbone of language understanding and the tools that fuel them.

Importance of the project

During this project, we will look at the process of sentiment analysis and use some of the latest cutting-edge algorithms to detect the sentiment of users after reviewing some of their favorite or least favorite movies. Sentiment analysis has many useful applications and use cases, and the reality is that many of these use cases are growing as the field develops. More time is being put into research, and compute power makes large scale analysis even easier. Some of the more common use cases for sentiment analysis include social media monitoring for companies that may want to understand how their customer base sees their products and services, customer support analysis to know where customers need to be directed to deal with their concerns over goods and services, reputation management to monitor how the general public or customer base views the organization and how it may be performing, product analysis, as well as other use cases. (*Sentiment Analysis Use Cases and Business Applications*, 2020)

Aims and Objectives

The aim of this project was to utilize transformer models to create a system that will accurately analyze text data to find the sentiment within the context of movie reviews. The sentiment can generally be positive or negative. While more detailed sentiment analysis is possible, the goal here was simply to see whether the model could accurately analyze comments into a binary evaluation of comments whether positive or negative. To that end, the output of this model will simply provide a 'positive' output or 'negative' output based on a comment posted in the text section. Additionally, we included a front-end endpoint to allow users to easily enter a text and have their comment analyzed at the click of a button. Another objective was to measure how well transformer models may have worked over other NLP solutions including Naive Bayes Classifiers or Support Vector Machine (SVM) models. The transformer of choice for this project was the DistilBERT Transformer, which improved accuracy over the other models previously mentioned. The motivation for utilizing a transformer model was due to the fact that transformers are the predecessor and foundation of Large Language Models (LLMs) which are now fueling such popular platforms as OpenAI's ChatGPT-4, Meta's Llama-2, and Google's BARD.

Problem Statement

As discussed previously, sentiment analysis can have many use cases. From those use cases are learning whether a customer is satisfied with a product or a service. In this case, we will analyze sentiment regarding movies based on descriptors used by the reviewer. We need to accurately assess user comments to later evaluate the value of a product, in this case, the value of a movie. Movie studios would be keen on knowing how movie-goers feel about a movie in order to produce others that may be enjoyed by their customer base. Many studios would be keen to invest heavily in such information that can be compiled by machine learning systems such as a DistilBERT model.

Scope of the Project

This project used more than 20,000 user comments split into a testing and training dataset for the purpose of training the transformer model on a fraction of the overall comments while setting aside a test set for testing the models accuracy. The dataset was created by Stanford University's AI Lab and is known as the 'Large Movie Review Dataset'. (*Sentiment Analysis*, n.d.)

Model Review

Before settling on the final model and evaluation metrics, there were two models that were considered for their overall accuracy and performance. Those two models included Naive Bayes Classifiers and Support Vector Machines. The Naive Bayes Classifier works by assuming that one feature in a class doesn't affect the presence of another. (Vadapalli, 2020) As Vadapalli states, "you'd consider fruit to be orange if it is round, orange, and is of around 3.5 inches in diameter. Now, even if these features require each other to exist, they all contribute independently to your assumption that this particular fruit is orange. That's why this algorithm has 'Naive' in its name."(Vadapalli, 2020) Therefore, features are independent of each other in a Naive Bayes Classifier.

While Naive Bayes Classifiers save time in training, require less training time, and are better suited for categorical inputs, it also has some limitations that encourage finding other solutions for this sentiment analysis project. (Vadapalli, 2020) There are limitations to the Naive Bayes Classifier that become glaringly clear once you look deeper into the algorithm. Naive Bayes assumes that all features are independent, however reality tells us that this is hardly the case in real-life scenarios. (Vadapalli, 2020) As Vadipalli also mentions, the Naive Bayes classifier assigns zero probability to a categorical variable that was not seen during the training phase of model building. (Vadapalli, 2020)

Given the limitations of the Naive Bayes Classifier, we examined another approach that could have provided the necessary solutions to the Naive Bayes Classifier problems. This approach was known as Support Vector Machines (SVMs). A Support Vector Machine works by trying to find the best decision boundary that separates the best data points of different data classes. (*What Is Support Vector Machine (SVM)? - Definition from WhatIs.Com*, n.d.) This works exceptionally well with binary classification problems like the one featured in this project. However, on closer inspection, we found that SVMs also have limitations. SVMs are not best suited for larger datasets nor are they suitable when there is class overlap, which happens often in data science projects. (K, 2020) Also, SVMs underperform when the number of data points exceeds the number of training samples. (K, 2020) Taking this into consideration, we had to find another solution that could offset the limitations of both Naive Bayes Classifiers and Support Vector Machines.

Consequently, our solution of using transformer models, in particular DistilBERT models, helped to overcome some of the setbacks of the previous two proposed solutions. Although it must be mentioned that there is no 100% effective machine learning solution, Transformers provide the best accuracy and performance to date outside of other Large Language Models (LLMs). In fact, Transformers are the foundation of Large Language Models. Furthermore, Transformers can understand the relationship between sequential elements that are far from each other. (Srivastava, 2022) They also pay very close attention to elements in a sequence and can train more data in less time than more traditional NLP-based machine learning models. (Srivastava, 2022) For these reasons, we decided to focus on the usage of transformer models to produce this project report and document how transformers ultimately were the better choice for evaluation of sentiment.

Methodology

To create our sentiment analysis model, we elected to use the Stanford AI Lab's Large Movie Review Dataset. This is a curated list of polarized movie reviews for training. There are 25,000 reviews for training and another 25, 000 for testing. (*Sentiment Analysis (Stanford AI Lab)*, n.d.)

In order to create models and perform exploratory data analysis, there were a few frameworks used to complete this process. The primary framework was PyTorch. According to PyTorch's own documentation, this framework is described as "...an optimized tensor library for deep learning using GPUs and CPUs." (PyTorch, 2019) Also, we used the DistilBERT algorithm via HuggingFace to serve the transformer model. DistilBERT is a smaller and faster distilled BERT model with 40% less parameters than normal BERT, but also performs 60% faster. (*DistilBERT*, n.d.) We used the Scikit-Learn framework to split the data into testing and training sets. Scikit-Learn is an open source library that bundles several different machine learning algorithms for the purpose of data analysis. (*What Is Scikit-Learn In Python?*, n.d.) However, Scikit-Learn was used here for its ability to easily split data into training, testing, or validation sets. Finally, we used Gradio to create a simple and easy-to-use front end to enter reviews and see the final analysis of each review whether negative or positive. Gradio is an easy to use and maintain app builder that does not require heavy front-end development. Using just a few short commands, Gradio can easily put together an application that has most of the essential functionality of a fully-built front-end app. (Team, n.d.)

The system design begins by importing all necessary libraries and frameworks. After this, there is a class featuring three separate functions that take in data and return a dataset that encodes the comments and labels them either with a positive or negative label. Then, there is another function that designates either a '1' for positive comments or '0' for negative comments and groups them with the labels created by the functions in the previously-mentioned class. Once this is completed, the data is loaded into two variables labeled 'train_texts' and 'train_labels' and 'test_texts' and 'test_labels' respectively. After this, 20% of the training data is set aside for tuning the model. Once this step has been completed, then the DistilBERT Tokenizer is initiated. Data is then tokenized into three categories, the first being train encodings, the second being validation encodings, and finally test encodings. Subsequently, these encodings and labels are converted into PyTorch datasets. These are the necessary steps to preprocess the data in order to start the training of the model. The model is then initialized and the training parameters set.

After model initialization and the trainer is created, then the process of training the model on the training data commences. There is a trainer evaluation function that then returns the best metrics that the model was trained on. Finally, there is a function that predicts on the test data based on the best metrics from the previous step of model training. Those metrics are then fed to a Gradio end-point that allows users to classify movie reviews at the touch of a button using Gradio's app interface. As stated previously, we used the DistilBERT Transformer due to its rapid training time despite being trained on millions of data points. (*DistilBERT*, n.d.)

Conclusion

While DistilBERT performed better than many traditional ML algorithms, there is still room for improvement in the area of accuracy. That said, DistiBERT still provides a robust transformer-based model that works extremely fast given the number of parameters it was trained on. It also outperforms traditional algorithms and gives developers a base to use that will eventually lead to more robust Generative AI solutions. This project is just a small taste of what sentiment analysis could become in the future. Here are some of the best metrics achieved during training:

{'eval_loss': 0.6953861713409424,

'eval_runtime': 2.0121,

'eval_samples_per_second': 0.994,

'eval_steps_per_second': 0.497,

'epoch': 50.0}

Better accuracy could have been achieved had the number of epochs increased or other hyperparameter tuning techniques used, however it would have created too much strain on the hardware used to conduct this project leading to either running out of sufficient memory or significantly increasing training time. Therefore, we had to choose simpler parameters to make sure the experiment could run completely.

Future Work

Moving forward, this work can be extended by experimenting with other transformer models. For example, we would like to train the same model with multiple transformers including GPT 3.5 or higher, XLNet, RoBERTa, ALBERT, StructBERT, DeBERTa, T5, UniLM, and Reformer.(Choudhury, 2021) Additionally, with new models such as LangChain and LLama-2 being produced, it would also be nice to see how these models perform comparatively.

In future contexts, perhaps sentiment analysis can evolve into AI Emotion. There has been a serious influx of computer science students hoping to harness the possibilities of AI becoming sentient, perhaps motivated by science fiction movies and entertainment. (Science, 2022) Also as compute power, storage, and data grows, we may be on the eve of such emotional intelligence by machines. At this point, the possibilities remain endless.

References

Choudhury, A. (2021). *Top Ten BERT Alternatives For NLU Projects*. Analytics India Magazine. https://analyticsindiamag.com/top-ten-bert-alternatives-for-nlu-projects/

DistilBERT. (n.d.). Huggingface.Co. https://huggingface.co/docs/transformers/model_doc/distilbert

K, D. (2020). *Top 4 advantages and disadvantages of Support Vector Machine or SVM*. Medium. https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107

PyTorch. (2019). *PyTorch documentation—PyTorch master documentation*. Pytorch.Org. https://pytorch.org/docs/stable/index.html

Science, O.-O. D. (2022). *The Evolution of AI Emotion and Sentiment Analysis*. Medium. https://odsc.medium.com/the-evolution-of-ai-emotion-and-sentiment-analysis-611b118bd983

Sentiment Analysis. (n.d.). Ai.Stanford.Edu. https://ai.stanford.edu/~amaas/data/sentiment/

Sentiment Analysis (Stanford AI Lab). (n.d.). Ai.Stanford.Edu. Retrieved January 1, 2023, from https://ai.stanford.edu/~amaas/data/sentiment/#:~:text=This%20is%20a%20dataset%20for

Sentiment Analysis Use Cases and Business Applications. (2020). MonkeyLearn Blog. https://monkeylearn.com/blog/sentiment-analysis-applications/

Srivastava, A. (2022). *What Are Transformers In NLP And It's Advantages*. Knoldus Blogs. https://blog.knoldus.com/what-are-transformers-in-nlp-and-its-advantages/

Team, G. (n.d.). *Quickstart*. Gradio.App. Retrieved January 1, 2023, from https://www.gradio.app/guides/quickstart

Vadapalli, P. (2020). *Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2020*. UpGrad Blog. https://www.upgrad.com/blog/naive-bayes-explained/

What is Scikit-Learn In Python? (n.d.). ActiveState. https://www.activestate.com/resources/quick-reads/what-is-scikit-learn-in-python/

What is support vector machine (SVM)? - Definition from WhatIs.com. (n.d.). WhatIs.Com. https://www.techtarget.com/whatis/definition/support-vector-machine-SVM