Sentiment Analysis of Indian News Headlines

1 Introduction

This report contains the IDS 703 final project for Aditya John, Scott Lai, and Pooja Kabber. We focus on comparing the performance of generative and discriminative NLP models on real world data versus artificially generated data. Since discriminative models focus on predicting the labels based on training data and generative models focus on explaining how the data was generated, we expect generative models to do better than the discriminative models on artificially generated text and vice versa with respect to real world natural language. To test this, we have chosen Naive Bayes to be our generative model and Bert to be our discriminative model. We will run this test on two datasets, one real world dataset consisting of Indian financial news headlines and a synthetic dataset generated by our Naive Bayes model.

2 Data

2.1 Description

The dataset for this project is the Indian Financial News Headlines dataset from kaggle. It consists of **200288 unique news headlines** with the dataset having **54% negative and 46% positive posts**. The dataset also contains confidence thresholds for the sentiment score it has assigned to either positive or negative sentiment.

For this project, we only use a subset of the dataset. First, we filter only to include articles with a confidence greater than 99%. Post which we shuffle the dataset and use 25k positive and negative samples. Our final dataset therefore contains of 50k datapoint with **50% positive and 50% negative posts**. Our reasons for doing this is twofold - first is to improve the quality of the dataset, by applying the confidence threshold. Additionally, by restricting the dataset to only **50k headlines**, we reduce the computational requirements for the discriminative model.

2.2 Pre-Processing

For the generative model, cleaning and pre-processing the dataset is important to improve the results. Because the model used is a bag-of-words model, reducing the noise in the dataset will help the model learn a more appropriate probability distribution.

We perform the following pre-processing steps

2.2.1 Stopword Removal:

Words that frequently occur in the corpus that dont convey any meaning are considered stopwords. For example, words like “the”, “and”, “an” etc. which dont have any inherent meaning but are required to form grammatical sentences are considered stopwords. Removing them for the corpus before training, will allow the model to focus more on words that contribute to the task of sentiment analysis.

2.2.2 Non-alphanumeric character removal:

This is performed to clean the dataset further. As our dataset is only headlines, we are not concerned with any punctuation marks or other non-alphanumber characters.

2.2.3. Stemming

Stemming is the process of reducing words in the sentence to their root stem. For example, for words like “larger” and “largest”, the root stem is “large”. Performing stemming on the dataset helps reduce the vocabulary as different inflections of the word get merged into one. Typically, lemmatization works better than stemming as it focuses on the “lemma” of the word rather than the “stem”. For example, after stemming the words “be”, “is” and “are” remain different, whereas when applying lemmatization, they get merged to the same word “be”. Lemmatization would further reduce the dataset as compared to stemming. However, our reason for stemming was primarily due to resource constrains as lemmatization is a lot more computationally intensive.

2.2.4. Removing Numbers

While this is not a usual pre-processing step, numbers do not add a lot of meaning when it comes to sentiment analysis. For example, consider the two headlines “Sensex , Nifty continue to struggle ; BHEL , Maruti rise 1 %” and ”2017 could be cyclically a very good year for largecap IT : Hiren Ved , Alchemy Capital”. In both cases, the numbers cannot be used to determine the sentiment score. However, it is possible that the numbers could bias the model towards one or the other sentiment. For example, most of the headlines during 2020 would be negative given the market crash and negative macro environment. Therefore, 2020 would have have a higher negative probability when compared to positive probability. Any sentence that has the number 2020 would be biased towards having a negative sentiment.

2.3 Synthetic Data Generation

Synthetic data refers to data created by the generative model. Post training, the model learns the posterior probability of the words for each class (positive and negative). By sampling the distribution of each class (positive and negative) we can generate word distributions of a given length. Because of the “naive” assumption of the model, i.e. that each word is independent of the other, each word is sampled independently. Therefore, it is very likely that the sentence generated do not follow any grammatical rules, but simply a set of words that follow the probability distribution.

3 Model

3.1 Generative

For sentiment analysis of text data, including news headlines, Naive Bayes is a generative model that can be used. Its foundation is the notion that predictions about the class (such as positive, negative, or neutral) of a given text can be made using probabilities. For sentiment analysis, news headlines are categorized as having a positive, negative, or neutral sentiment, and this information is used to train the model. The algorithm learns the likelihood of various words or phrases occurring in headlines with each sentiment class using this training data. The model determines the likelihood that a given headline belongs to each sentiment class during the prediction phase based on the words and phrases it includes. The sentiment of the headline is then predicted to be the class with the highest likelihood. The "naive" assumption, one of the fundamental elements of the naive Bayes model, states that all features (in this case, the words and phrases in the headline) are independent of one another. This presumption makes the computations easier and makes it possible to train and operate the model quickly, but it might not always be true in actual use. Overall, naive Bayes is a straightforward and efficient technique for doing sentiment analysis, and it has been widely applied for this purpose in a variety of scenarios, including the analysis of headlines from Indian news sources.

3.2 Discriminative

For the discriminative model, we are approached by using the BERT approach to do the sentimental modeling analysis. BERT is a transformer-based model that has shown amazing results in various NLP tasks over the past years. We are fine-tuning the model using the Hugging face and applying the pre-train model in our ESG test set. A popular machine learning approach for natural language processing applications like language translation, text classification, and text synthesis is called BERT (Bidirectional Encoder Representations from Transformers). It is a discriminative model that has been trained to forecast a particular outcome based on input data, such as the class or label of a piece of text (e.g. the words and sentences in a document). We initially compiled a collection of news articles about India from Indian Financial News Headlines and categorized them with pertinent categories or tags before using BERT to assess the news in India. Politics, economy, culture, and other topics are how we categorize articles. We utilize BERT to train a model to categorize new articles based on their content after generating the dataset. We feed the news items as input data and the matching output labels to the model to train it (the categories or tags). We optimize the model's internal parameters to reduce the discrepancy between the anticipated and actual labels. Once the model has been trained, we feed new articles into it and use the model to forecast the most likely label based on the article's content.

4. Results

4.1 Generative Model

On the real data, we observe an accuracy of **87.16%**. Comparatively, the model was less computationally intensive compared to BERT, not requiring a lot of time to train with the only major requirement being memory to hold the large document vectors in memory. The interpretability of the model is quite high, as we are able to vizualize the underlying probabiltiy distribution for both the positive and negative sentiment class. Therefore, when we get a classification result, we can simply refer to the distribution to understand how (or why) a particular headline was classified with a certain sentiment.

On the synthetic data, we observe an accuracy of **91.28%**. As the data was generated from a generative model itself, it follows that the data generated would adhere to the underlying distribution, explaining the rise in accuracy.

4.2 Discriminative Model

On the real data, we observe an accuracy of **92.01%**. The model was more computationally intensive both in terms of time and resources, a GPU was required for fine-tuning the model. The interpretability of the model is quite low, as it is not possible to intutively understand why a particular document is getting classified as either negative or positive. While it is possible, to mathematically explain how the model works, it is not something that would be intutive to understand.

On the systhetic data, we observe an accuray of **49.40%**. This again, follows our expectations as the data generated does not confine to grammatical rules and is simply a set of words generated by the model. BERT is not a bag-of-words model, having been trained on millions of grammatical sentences. Therefore, when being fine-tuned on a dataset generated by a bag-of-words model, it is not able to generalize well and therefore leads to poor performance. We observe an accuracy of less than 50%, which is essentially the same as flipping a coin to determine the sentiment.

5 Conclusion

From this project, we clearly understand the differences between a generative and discriminative model and when one is preffered over the other. Generative models are typically used when interpretibility is important, and we are not as concerned with the accuracy of the model. On the other hand, a discriminative model clearly has better performance on real world data as evidenced by the higher accuracy. But the increased accuracy comes at a loss of interpretibility.