IDS703 Final Project Report

Anna Dai, Satvik Kishore, Moritz Wilksch

December 12th, 2021

1 Introduction

In this project, we have worked on tweet classification as a Natural Language Processing problem, more specifically, as a document classification problem. Twitter is a microblogging service where users post publicly visibile "tweets", that are essentially texts with less than 280 characters. These tweets may also contain other media objects which are discarded for the purposes of this project. These tweets most often serve as discussion piceses as part of larger conversations. They are relevant to any number of topics under discussion. These "topics" are also often explicilty highlighted by the user using a "hashtag", i.e. text with '#' followed by the topic name, or a commonly used shorthand for it. In our project, we treat these hashtags as "topics" for our document classification model, where each tweet is an instance of a document.

2 Data

We have manually selected 7 topics, or hastags for classification. These are:

- crypto
- tesla
- GSW
- formula1
- thanksgiving
- holidays
- covid19

These topics were intentionally selected to have topics that have some amount of overlap between them (holidays and thanksgiving) while also having topics easier to differentiate (crypto vs formula1). We scraped data using the python library twint [**?**], scraping approzimately 10,000 tweets for each of the seven topics. We also applied a few pre-processing steps before the next steps. This included tokenization of the tweets into words. ¡elimination of common words?¿. Conversion of emojis into tokens (each appearance of an emoji is a token. Multiple emojis strung together are treated as different tokens in sequence). We also removed punctuation marks. Following these steps, we split our data into three parts using a 60:20:20 split to form a training dataset, a validation dataset, and a test dataset.

3 Methodology

3.1 Generative Model

We used a Latent Dirichlet allocation (LDA) model as a generative model to learn from the corpus we have collected. This type of model does not require any hyperparamter tuning, and thus was trained using a combination of the training and validation dataset. We used the LDA implementaion from scikit-learn [**?**].

3.2 Discriminative Model

- network description - hyperparameter tuning on synth data - application to real - transfer learning on real data

- compare to model that has ONLY been trained on real?

4 Results

4.1 Benchmarking on Synthetic Data

4.2 Benchmarking on Real Data

5 Conclusion