Large Scale Sentiment Analysis of Tweets

James Tan Juan Whei
Concordia University

August 3, 2022

Abstract

Abstract, here is what an abstract is compared to an intro https://www.discoverphds.com/blog/abstract-vs-introduction

1 Introduction

An idea of what the project is about and its possible use cases?

2 Tools and Technologies

An introduction to the tools that were used and why they were chosen for this project

2.1 GCS

Some description about GCS

2.2 BigQuery

Some description about BigQuery

2.3 Dataproc and Spark

Some description about Dataproc and Spark. Mention cluster configuration, number instances, type of instances etc. Refer readers to our scripts for specifics.

3 Dataset

Some details about the dataset and how it was obtained.

4 Processing of Data

As mentioned in Section 2.3, we define the processing of the data as a Spark job. The steps involved in the job are illustrated in Figure 1 and will be further elaborated on in the subsequent sections. Our processing pipeline relies heavily on the Spark NLP library[5].

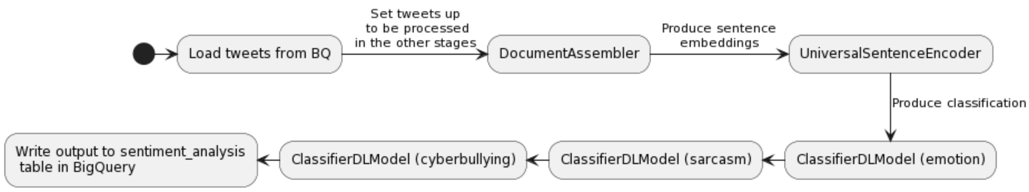

Figure 1: The processing pipeline

4.1 Data Ingestion

In order to use data stored in BigQuery as an input to our Spark job, we used the Spark BigQuery connector[2]. The Spark script reads from a table that contains all the tweets that were procured as described in Section 3.

Each run of the Spark job would typically be executed on 4-5 days worth of tweets as we discovered that the Spark jobs had a tendency of failing when working with larger amounts of data. This was true even when the CPU and memory utilisation of the worker nodes were relatively healthy and thus should be further investigated.

4.2 Document Assembler

The first step of the pipeline is the `DocumentAssembler`[4]. This prepares the data into a format that is processable by Spark NLP and is essentially the entry point for every Spark NLP pipeline.

4.3 Generation of Sentence Embeddings

We generate sentence embeddings by leveraging a Universal Sentence Encoder[6] made available by Tensorflow. The output of this stage is a 512-dimensional vector that semantically captures the meaning of each tweet. This is the basis upon which the downstream classification algorithms build on.

4.4 Sentiment Classification

To actually use the embeddings described in the previous section, we utilise ClassifierDLModels[3] to classify the tweets. Each ClassifierDLModels essentially assigns a label to each tweet. To identify the emotion, presence of cyberbullying and presence of racism in each tweet, we use the `classifierdl_use_emotion`, `classifierdl_use_cyberbullying` and `classifierdl_use_sarcasm` pretrained models respectively (TODO: Better way of phrasing this?).

The emotion classifier produces the values `sadness`, `joy`, `love`, `anger`, `fear` and `surprise`. The cyberbullying classifier produces the values `neutral`, `racism` and `sexism`. The sarcasm classifier produces the values `sarcasm` and `normal`.

4.5 Storing of Output

The output is then stored in a separate table in BigQuery. Note that the BigQuery Spark connector is once again used here, thus allowing the output of a Spark job to be appended directly to a BigQuery table.

5 Results

Some visualisations and the insights we obtained

6 Future Work

An idea that we originally had was to run an unsupervised clustering algorithm on the sentence embeddings produced in Section 4.3. The clustering algorithm that we had in mind was DBScan[1], with the objective of clustering tweets with similar topics. Unfortunately, we were not able to find a satisfactorily efficient implementation of the algorithm to employ with Spark. Due to the size of our dataset and the high number of dimensions of the embeddings, an efficient implementation was crucial to the success of this idea. Given that we had limited resources, we were forced to abandon this idea, however, it should be revisited in the future by implementing our own version of the DBScan algorithm.

7 Conclusion

We worked hard, and achieved very little.

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[2] Google. Use the BigQuery connector with Spark. https://cloud.google.com/dataproc/docs/tutorials/bigquery-connector-spark-example, 2022. [Online; accessed 3-August-2022].

[3] J. S. Labs. ClassifierDLModel. https://nlp.johnsnowlabs.com/api/com/johnsnowlabs/nlp/annotators/classifier/dl/ClassifierDLModel.html, 2022. [Online; accessed 3-August-2022].

[4] J. S. Labs. DocumentAssembler. https://nlp.johnsnowlabs.com/api/com/johnsnowlabs/nlp/DocumentAssembler, 2022. [Online; accessed 3-August-2022].

[5] J. S. Labs. Spark NLP: State of the Art Natural Language Processing. https://nlp.johnsnowlabs.com, 2022. [Online; accessed 3-August-2022].

[6] Tensorflow. Universal Sentence Encoder. https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder?hl=en, 2022. [Online; accessed 3-August-2022].