Sentiment analysis with tweets

Mohamed BENMAIZA, Christian KAYO

1 Context

We propose to conduct a project on sentiment analysis of Tweets. The primary objective of this study is to develop a model for sentiment analysis of tweets that can accurately classify the polarity of tweets as positive or negative. The proposed research project will focus on the following questions:

- How does the performance of the sentiment analysis model vary depending on the choice of word normalization technique ?
- How does the sentiment of tweets vary with different embedding techniques ?
- What are the most effective neural networks for sentiment analysis of tweets?
- How do sequence models compare with attention based models?

To answer these questions, we will collect a large dataset of tweets called sentiment140 from kaggle. We will preprocess the tweet data by removing stop words, applying stemming or lemmatization and tokenizing. Then, we will use various word embeddings, to convert the tweet data into numerical representations. We will compare the performance of various deep learning learning algorithms for sentiment analysis of tweets.

2 Dataset

We this project, we used the sentiment140 dataset whiich is a widely used dataset in the field of sentiment analysis. It contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated with a polarity score of either 0, indicating a negative sentiment, to 4, indicating a positive sentiment.

The dataset consists of six fields including :

- the polarity score,
- the id of the tweet,
- the date of the tweet,
- the query used to extract the tweet (if any),
- the user who tweeted the message,
- the text of the tweet itself.

The dataset was first released in 2009 and has since been widely used in research and academia. The sentiment140 dataset provides a valuable resource for researchers and practitioners, enabling them to develop and evaluate models that can accurately detect the sentiment of a given tweet.

3 Sequence models

Our architecture is made up of the following layers

- Embedding Layer - Generates Embedding Vector for each input sequence.
- Conv1D Layer - Its using to convolve data into smaller feature vectors.
- RNN or GRU or LSTM - a sequence layer.
- Dense - Fully Connected Layers for classification

3.1 Experiment: Stemming vs Lemmatization

Stemming reduces words to their base or root form, while lemmatization converts words to their dictionary form. By comparing these two models, we aimed to evaluate whether stemming or lemmatization is more effective in capturing sentiment information from tweets. So we think the choice of word normalization technique, whether stemming or lemmatization, can impact the performance of sentiment analysis models.

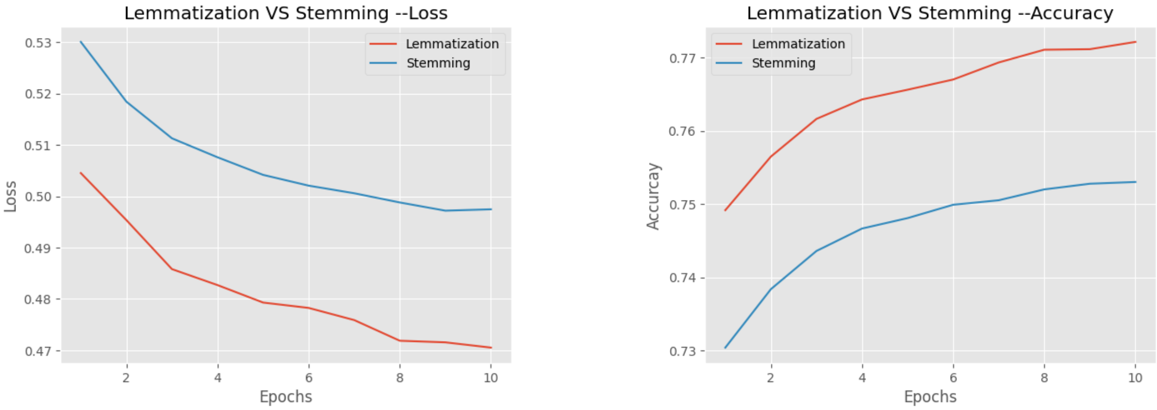

Figure 1: Results of the Stemming vs lemmatization experiment

Lemmatization performed slightly better than stemming on our data set , this is because lemmatization produces words that are closer to their original form and can provide better results in identifying sentiment. So for all our experiments we decided to use lemmatization over stemming.

3.2 Experiment: GloVe vs Word2Vec

By comparing these two models, we aimed to evaluate the impact of different word embedding techniques, Word2Vec and GloVe, on the sentiment analysis performance. Word2Vec represents words as dense vectors based on the context in which they appear, whereas GloVe word embeddings are derived from global word co-occurrence statistics. Comparing these models allows us to determine which word embedding technique is more effective for sentiment analysis on our dataset.

Word2Vec performed slightly better than GloVe on our data set. So for our next experiment we decided to use Word2vec over GloVe.

3.3 Experiment: RNN vs GRU vs LSTM

We aimed to compare the performance of three different sequence models, namely LSTM (Long Short-Term Memory), SimpleRNN (Simple Recurrent Neural Network) and GRU (Gated Recurrent Unit) for sentiment analysis using Word2Vec word embeddings. The objective was to investigate the impact of different sequence models on the sentiment analysis task.

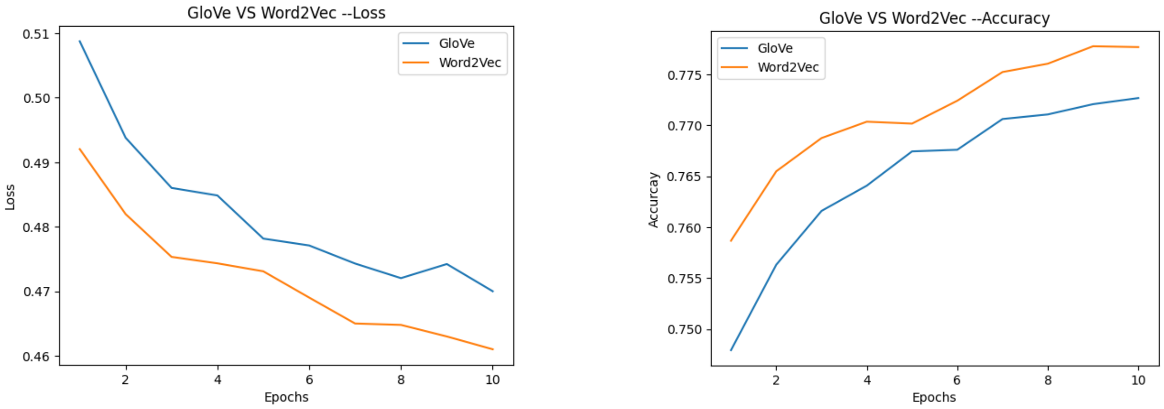

Figure 2: Results of the GloVe Vs Word2Vec experiment

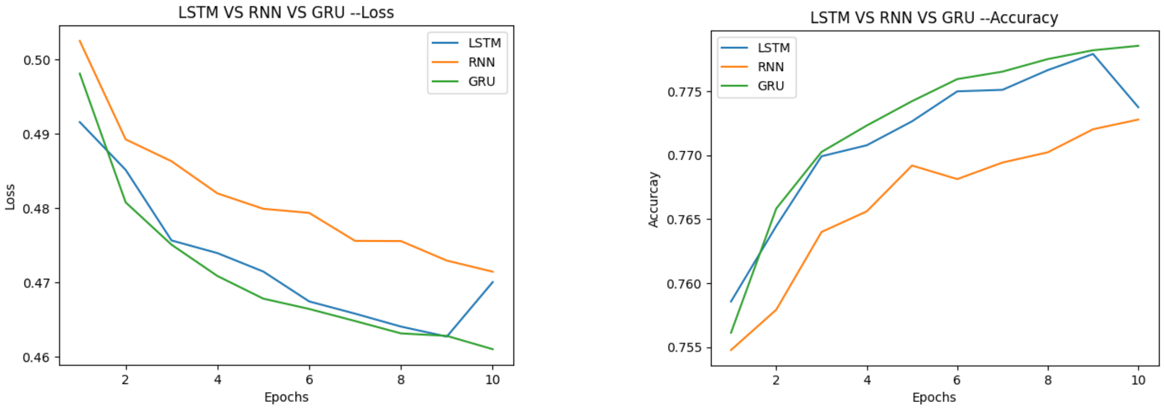

Figure 3: Results of the RNN vs GRU vs LSTM experiment

As we expected GRU and LSTM performed better than the base line RNN. With GRU performing better than LSTM.

4 Attention based model

Our architecture is made up of the following layers

- Preprocessing Layer: Tokenizes and encodes the input text using the preprocessing capabilities of BERT.
- Encoder Layer: Uses a pre-trained BERT model to generate contextualized representations of the input text.
- Dense Layer: Performs a linear transformation on the BERT-encoded representation, increasing the model's expressiveness.
- Batch Normalization Layer: Helps the network training and regularization.
- Dropout Layer: It randomly disables a fraction of the dense layer's units to prevent overfitting.
- Output Layer: Generates a binary sentiment prediction (positive or negative) using a dense layer with a sigmoid activation function.

4.1 Comparing the based sequence model and Attention based model

We wanted to compare our best sequence based model with the best attention based model but unfortunately we could not run train the attention based model due to lack of computational resources.