University of British Columbia

Advanced Machine Learning Tools for Engineers

EECE 571T

Social Media Sentiment Analysis using LSTM - Group 18

Author: Zhaohang Yan

Author: Vicky Zhao

April 2023

Project Report: Social Media Sentiment Analysis using LSTM

1) **Goal of Report:** What is the goal of your project? (3 to 5 sentences)

   We propose the research to investigate the potential of machine learning for social media sentiment extraction and analysis, with the goal of developing models that can accurately and efficiently analyze social media data to extract insights about public opinion. The study will focus on creating novel sentiment analysis methods and analyzing how well they perform when applied to actual social media data. The results of this study will expand the field of sentiment analysis and offer valuable insights for businesses and decision-makers.

2) **Previous work done:** Here for one page, outline, using references, how this type of work was done, or work close to the current report subject was done (1 page max). **worth 10%**

   In recent years, the use of sentiment analysis has rapidly grown and become an increasingly important tool for organizations, governments, and individuals, due to its ability to quickly analyze vast amounts of social media data. Sentiment analysis offers valuable insights into public opinion and sentiment that can inform decision-making and lead to improved outcomes.

   In times of crisis, such as pandemics and natural disasters, sentiment analysis can offer useful information to public health experts and other organizations. For instance, according to the paper '*Deep Learning Reveals Patterns of Diverse and Changing Sentiments Towards COVID-19 Vaccines Based on 11 Million Tweets*,' sentiment analysis can be used to understand the evolving perceptions of COVID-19 vaccines, and reveal how the user characteristics interact with vaccinations [1]. Sentiment analysis can also be employed to investigate public attitudes towards a certain product using large-scale social media data. For example, '*Investigating The Impacting Factors on The Public's Attitudes Towards Autonomous Vehicles Using Sentiment Analysis from Social Media Data*' shows that sentiment analysis can be used to investigate public attitudes towards self-driving cars on social media [2]. This research result is conducive to the development of autonomous driving technology, automated guidelines for driving-related policy development, and public understanding and acceptance of autonomous driving.

   However, there are challenges and potential issues in studying sentiment analysis in social media that should be noticed by scholars and researchers. For example, the paper '*A systematic review of social media-based sentiment analysis: Emerging trends and challenges*' summarized challenges and issues of existing sentiment analysis research into three main categories: Data-related, Method-related, and Evaluation metrics [3]. The paper provides information about the objectives of the sentiment analysis task, the implementation procedure, and the ways it is applied in different application domains. According to the review '*Sentiment Analysis in Social Media and Its Application: Systematic Literature Review*,' there are two main methods of sentiment analysis that have been established, which are machine learning method and lexicon-based method [4]. It concludes choosing the appropriate method depends on the time and amount of data and text structure. The paper '*Exploration of social media for sentiment analysis using deep learning*' proposes sentiment analysis using deep learning methods and demonstrates decent accuracy performance [5].

3) **Strategy:** Your strategy for achieving your Goal – (half a page max). Include a short pseudo description of the final method used. **worth 15%**

There are three main steps in our strategy to achieve our goal:

Data Collection & Pre-processing: We used real data collected from a social media platform and pre-processed the dataset by dropping useless columns, decoding/encoding labels, and tokenizing.

Text Representation: We used word embedding methods to numerically represent the unstructured text documents and make them mathematically computable.

Model Training & Evaluation: We built a sequence model consisting of several layers to handle a sequence of data and learn a pattern of input sequence that captures the semantic relationships between words. To evaluate the model's performance, we analyzed its confusion matrix, accuracy, and training loss.

4) **What Methods Worked:** In more detail explain what worked. Maximum 4 and a half pages including figures as needed **worth 40% total.**

a. **Pseudo-code description** (**worth 10%**):

(a) Step 1: Data Collection & Pre-processing

i. Data Collection: We choose to use the *Large Yelp Review Dataset*, which is a dataset for binary sentiment classification. With 560,000 reviews for training and 38,000 for testing. It is a substantial dataset that should provide a good amount of variability and diversity in the reviews.

ii. Data Pre-processing: We drop useless columns and rename columns for training convenience. Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. The process is called Tokenization.

(b) Step 2: Text Representation

i. Word Embedding: Word embedding is a technique used in natural language processing (NLP) to represent words as dense vectors of real numbers, where each dimension of the vector corresponds to a certain feature of the word. In the project, we use the pre-trained Twitter GloVe Embedding package from Stanford AI.

(c) Step 3: Model Training & Evaluation

i. Architecture of Training: We use Sequence models with Long Short Term Memory (LSTM) to train the model. And there are four kinds of layers in our model:

1) First is the Embedding Layer, and it generates an Embedding Vector for each input sequence.
2) Second is the Convolution 1D Layer, we use this to convolve data into smaller feature vectors.

3) Third is the Long Short Term Memory(LSTM), which is a variant of RNN, and it has memory state cells to learn the context of the words which are further along the text to carry contextual meaning rather than just neighboring words.
4) Last we use Dense, which are fully Connected Layers for classification.

ii. Model Evaluation: We calculate the model's accuracy and model's loss and plot the graphs. We also use a confusion matrix to evaluate the performance of a model.

1) The accuracy is the proportion of correct predictions over the total number of predictions, while the loss is a measure of how well the model is fitting the data.
2) A confusion matrix is a table that shows the number of true positives, false positives, true negatives, and false negatives in a binary classification problem. The confusion matrix can be used to calculate various evaluation metrics such as precision, recall, and F1-score. Overall, the confusion matrix is a useful tool for evaluating the performance of a model, especially in binary classification problems. It provides a clear picture of how well the model is performing and can help identify areas for improvement.

b. **Explanation of Method Used and results**(**worth 30%**): Please explain why you chose the method that you used, justify the choice of the method over other possibilities including the evolution of your choice until you get to the final choice of method. Here you can connect your method choice to the data that you are using. Figures will come here.

(a) Step 1: Data Collection & Pre-processing

i. We use the Large Yelp Review Dataset because of several reasons.

First, the size of this dataset is very large, and it contains 560,000 reviews for training and 38,000 reviews for testing, which makes it one of the largest publicly available datasets for sentiment analysis.

Second, the data in this dataset is very diverse. The reviews in the dataset cover a wide range of topics and come from a diverse group of reviewers. This diversity helps ensure that the trained model is capable of generalizing to new data.

Third, as I mentioned earlier, the reviews in this dataset are highly polar, which means they are either overwhelmingly positive or negative. This makes it easier to distinguish between positive and negative sentiment when training a model.

Last but not least, the reviews in the dataset come from Yelp, a popular review website currently. This means the sentiment analysis task is directly relevant to a real-world application, which can be useful for evaluating the performance of the model.

The *Large Yelp Review Dataset* is better for binary sentiment classification tasks, as it is labeled with positive or negative sentiment, while the *Sentiment140 dataset* with 1.6 million tweets indicates whether each tweet is positive, negative, or neutral, it is more suited for multiclass classification tasks. Thus we change to use the Yelp dataset.

ii. We use tokenization in NLP tasks for several reasons.

First, tokenization is usually the first step in pre-processing data for NLP tasks. Once the

text has been tokenized, we can perform other pre-processing steps such as removing stopwords, or lemmatization words.

Second, in order to represent text data as numerical features which can be used as input to the model, we need to convert the text into a sequence of tokens. These tokens can be encoded using techniques like word embeddings after.

Third, tokenization can help us to improve the computational efficiency of NLP tasks by reducing the size of the input data. By breaking down a piece of text into individual tokens, we can represent the text more efficiently and avoid processing unnecessary information.

(b) Step 2: Text Representation

Word embedding is often used in NLP tasks because it provides a way to represent words as dense vectors of real numbers, which can be easily processed by CNN or RNN algorithms.

Word embedding allows us to represent words in a lower-dimensional space than their original one-hot vector representation. This makes it easier to process the data and can lead to better model performance.

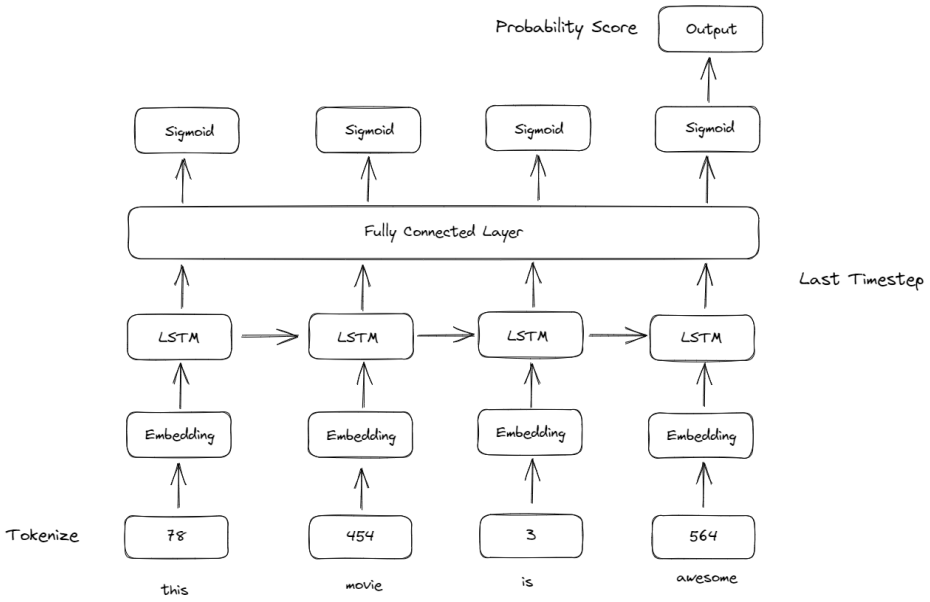
Additionally, word embeddings can also capture the semantic relationships between words. Words that are semantically similar will have similar vector representations, which can be useful in downstream NLP tasks such as sentiment analysis, text classification, and machine translation.

Last but not least, Word embeddings can capture contextual information about words, which can be helpful and useful in the tasks that require the comprehension of the meaning of words in context.

In our project, we choose to use pre-trained *Twitter GloVe Embedding* from Stanford AI, because pre-trained embeddings have already been trained on large amounts of text data, which can lead to improved performance on downstream NLP tasks. Instead of having to train our own embeddings from scratch, we may use pre-trained embeddings to take advantage of the knowledge that has been learnt from a sizable corpus of text. In addition, training word embeddings can be a computationally intensive process, especially when working on large amounts of data. By using pre-trained embeddings, we do not need to train our own embeddings, which can save time and computational resources.

(c) Step 3: Model Training & Evaluation

i. Architecture of Training: We use the sequence model with Long Short Term Memory (LSTM) to train the model, because they are able to capture the sequential dependencies between words in a text document.



Unlike traditional feedforward neural networks that process fixed-size inputs, sequence models can take variable-length sequences of inputs, such as the words in a sentence and the characters in a word, and produce corresponding outputs. LSTM is a type of RNN which is designed to overcome the vanishing gradient problem that can occur when training RNNs on long sequences. LSTM uses a memory cell which allows them to selectively remember or forget information from previous timesteps in the sequence. This memory cell can store information for an extended period of time, making LSTMs particularly useful for capturing long-term dependencies in a sequence.

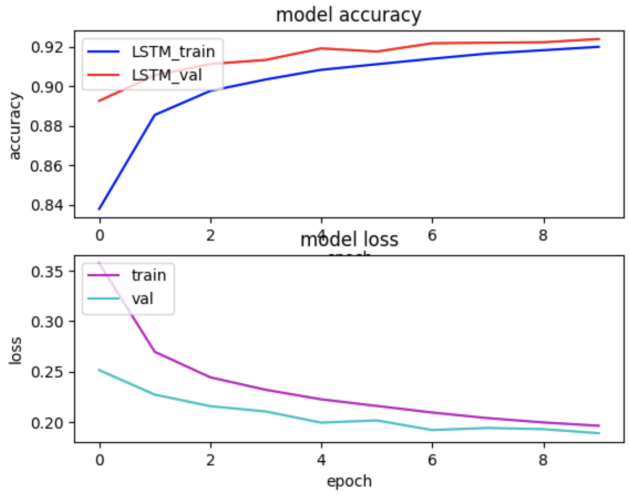
There are 4 kinds of layers in our sequence model:

Embedding Layer: This layer creates an embedding vector for each word from a sequence of integer indices that represent the words in a text document as input. The word's meaning and context in the document are captured by a detailed representation of the term called an embedding vector. Typically, pre-trained word embeddings, like GloVe embeddings, are used to initialize the embedding layer.

Convolution 1D Layer: The embedding layer's sequence of embedding vectors is subjected to a 1-dimensional convolution operation in this layer. By convolution over smaller portions of the sequence to create smaller feature vectors, this layer aims to extract features from the embedding sequence.

LSTM Layer: This layer is a variant of a Recurrent Neural Network (RNN), which is designed to capture long-term dependencies between words in a sequence. Memory state cells in the LSTM layer enable it to recall data from earlier in the sequence and carry it over to subsequent portions of the sequence. By learning the context of words rather than just their near neighbors, the model is able to understand their meaning.

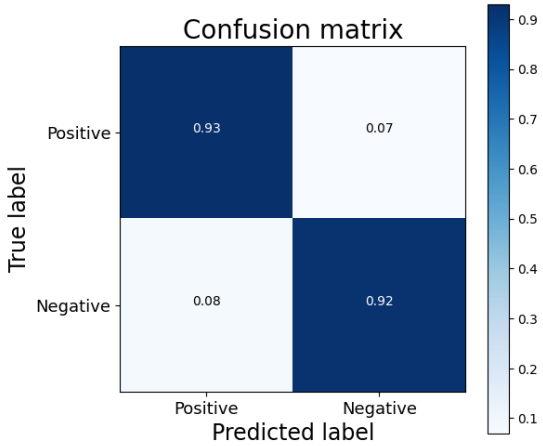
Dense Layer: This fully connected layer uses the LSTM layer's output as an input and generates a classification output. This layer's function is to convert the series of LSTM outputs into a final classification output, such as a binary classification (for example, a positive or negative sentiment classification) or a multi-class classification (for example, a topic classification).



ii. Model Evaluation:

A frequent statistic for assessing a machine learning model's performance is accuracy. It calculates the percentage of accurate predictions the model makes compared to all other forecasts. While accuracy is a crucial criterion for assessing a model's performance, it can occasionally be deceptive, particularly when the data is unbalanced. In these situations, the model may perform well on the majority class while underperforming on the minority class, producing a high overall accuracy while underperforming on the minority class. Other metrics, like precision, recall, and F1 score, can be utilized to get around this issue.

On the other hand, loss is a measure of how well the model is fitting the data during training. A machine learning model is trained with the intention of minimizing the loss function, a mathematical function that assesses the discrepancy between the model's expected and actual outputs. The categorical cross-entropy loss is the most widely used loss function for classification issues. The better the model fits the data, the lesser the loss.



In machine learning, a confusion matrix is a table that summarizes the performance of a classification model on a set of test data. To evaluate the performance of our sentiment analysis model, we can calculate the number of correct and incorrect predictions for each sentiment class. A confusion matrix can help us visualize it by showing the values of true positive, true negative, false positive, and false negative predictions made by the model for each sentiment class.

5) **What did not work. [**max 1 and a half pages**] worth 20%**

Describe what did not work in achieving the results in **4b**.

Data Collection & Pre-processing:

At first, we considered using the *Sentiment140 dataset*, which contains 1,600,000 tweets. However, we ultimately decided to use the *Large Yelp Review Dataset*, which has around 600,000 reviews on Yelp. While both datasets are collected from social media platforms, the Sentiment140 dataset was collected in 2009, while the Yelp dataset was collected in 2015. We felt that the Yelp dataset would provide a more accurate reflection of people's linguistic habits and context on modern social media.

Text Representation:

Initially, we considered using feature extraction to represent our text data in a machine-readable format. However, we eventually decided to use word embedding for text representation. Both feature extraction and word embedding are techniques used in natural language processing (NLP) to represent text data in a numerical format that can be processed by machine learning algorithms.

Feature extraction involves identifying and selecting relevant characteristics or features from raw text data. These features can include word frequency, sentence length, and part-of-speech tags. Feature extraction is often used to reduce the dimensionality of the data and make it easier to process.

On the other hand, word embedding is a technique for representing words as vectors of numerical values. This is achieved by mapping each word to a high-dimensional vector that captures its semantic meaning based on its context in a large corpus of text. Word embeddings are widely used in NLP tasks, such as language modeling, sentiment analysis, and text classification.

The main reason why we decided to switch to word embedding is that traditional feature extraction may struggle to accurately capture sentiment in ironic and sarcastic statements in social media.

Consider the tweet "*Great, just what I needed. Another meeting to attend. #not*". Although the words "great" and "attend" may appear positive, the hashtag "not" indicates a negative sentiment. Traditional feature extraction techniques that rely on word frequency and other similar metrics may not be able to account for this level of complexity in language usage.

On the other hand, word embeddings have the ability to capture the semantic meaning and context of words, enabling a more nuanced sentiment analysis that can account for such irony. By analyzing the entire context of a sentence or document, word embeddings can provide a more complete picture of the sentiment being expressed, which is especially important in social media where sarcasm, irony, and other forms of figurative language are often used.

Pre-trained Word Embedding vectors:

We also made a switch in our pre-trained word embedding vectors from *glove.6B* (Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 300d vectors) to *glove.twitter.27B*, which was specifically trained using data from Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 200d vectors). This decision aligns well with the objectives of our project, which is to study sentiments in social media. With a word embedding model trained on Twitter data, we can capture the nuances of social media language more accurately and create more contextually relevant word embeddings,

which can help improve the performance of our sentiment analysis model on real social media data.

6) Conclusions and Future work - 1/2 page, worth 5%

In conclusion, our project successfully completed accurate (around 93% accuracy) sentiment analysis for two general sentiments, which are positive and negative.

In the future, we can expand our analysis by incorporating more sentiment categories such as neutral, happy, angry, excited, disappointed, fearful, surprised, and so on. This will allow us to provide a more comprehensive and detailed sentiment analysis of social media data and capture the nuances of public opinions.

Furthermore, besides completing sentiment analysis for diverse sentiments, we are also considering building automated bots on social media platforms to react to public opinion regarding particular topics. One of our concepts is to create a bot that can monitor cryptocurrency market sentiment and make trades automatically based on that sentiment.

By scrutinizing social media data and public sentiment, the bot can recognize patterns and fluctuations in market sentiment and quickly adjust the trading strategies to them. Specifically, the bot can assist traders or investors in making more informed decisions based on the current market sentiment, which can result in higher profits and better risk management.

Though we used LSTM and sequence models in our project, in future work, we can investigate other deep learning and machine learning models, such as the Transformer model, to evaluate their performance in social media sentiment analysis. We will assess their performance based on multiple metrics, including accuracy, difficulty, and training efficiency, in order to identify the optimal model for future projects.

7) References

1. Deep Learning Reveals Patterns of Diverse and Changing Sentiments Towards COVID-19 Vaccines Based on 11 Million Tweets
   https://arxiv.org/pdf/2207.10641.pdf
2. Investigating The Impacting Factors on The Public's Attitudes Towards Autonomous Vehicles Using Sentiment Analysis from Social Media Data
   https://arxiv.org/pdf/2108.03206.pdf
3. A systematic review of social media-based sentiment analysis: Emerging trends and challenges
   https://www.sciencedirect.com/science/article/pii/S2772662222000273
4. Sentiment Analysis in Social Media and Its Application: Systematic Literature Review
   https://www.sciencedirect.com/science/article/pii/S187705091931885X
5. Exploration of social media for sentiment analysis using deep learning
   https://link.springer.com/article/10.1007/s00500-019-04402-8