Twitter Sentiment Analysis

1st Jun Du 2nd Baofeng Wang 3rd Yutong Wang 4th Isla Zhu 5th Shulin Li

I. Introduction

Twitter is one of the most popular social platforms today where people share their views and opinions on various events. Sentiment analysis can measure customer satisfaction and help companies to better improve their products. To make a better sentiment analysis, they usually extract information from social platforms like Google, Facebook, and Twitter.

Natural Language Processing (NLP) plays a vital role in the field of Twitter Sentiment Analysis. NLP technology can help in pre-processing by cleaning the text, punctuation and eliminating special characters. For example, removing stop words such as ”I”, ”my”, ”our”, ”was” and ”is”.

The objective of this project is to develop a sentiment analyzer that addresses the challenges of accurately identifying the sentiments (positive or negative) within Twitter tweets. The proposed solution involves implementing a neural network using the TensorFlow framework. We will use three methods: machine learning, deep learning, and a combination of machine learning and deep learning simultaneously, to compare and analyze the most suitable model for the sentiment analyzer.

II. Literature Review

A. EDA

EDA is an acronym for ”Exploratory Data Analysis.” It's a phase within data analysis where data analysts or scientists visually and statistically delve into a dataset to condense its primary attributes. The objective of EDA is to acquire a deeper understanding of the data and unveil patterns, connections, irregularities, and trends that can provide direction for subsequent analysis and decision-making.[1] Initially formulated by John Tukey (an American mathematician) during the 1970s, the EDA techniques persist as a widely employed approach in contemporary data exploration practices.[2]

B. Machine Learning

Machine learning constitutes a facet of artificial intelligence (AI) and computer science that centers on employing data and algorithms to simulate human learning, progressively enhancing its precision.[2] In the inception of the AI field during the 1950s, AI was characterized as the capability of any machine to accomplish tasks that typically necessitate human intelligence.[3] In this project, we will use a random forest algorithm to analyze Twitter sentiment.

As machine learning technology has evolved, it undeniably streamlined our daily lives. Incorporating machine learning into enterprises has also prompted a range of ethical considerations surrounding AI technologies. On the one hand, while this subject captures significant public interest, numerous researchers do not hold apprehensions about the prospect of AI outstripping human intelligence in the immediate future. [3]

C. Deep Learning

Deep learning involves a neural network with at least three layers, which learns from vast amounts of data and aims to mimic the human brain as closely as possible.

In the 1920s, Wilhelm Lenz and Ernst Ising formulated and analyzed the Ising model[4], which represents an inherent non-learning RNN architecture composed of neuron-like threshold elements. Subsequently, in 1972, Shun'ichi Amari introduced adaptability to this structure[5], later popularized as a learning RNN by John Hopfield in 1982[6]. By the late 2000s, deep learning commenced its ascendancy over alternative methods in machine learning competitions. In 2015, CNNs surpassed humans for the first time in an object recognition challenge. In our project, we will use CNNs to develop a sentiment analyzer.

III. Model

In this project, we will use two methods to perform sentiment analysis. The first method involves machine learning, where we will utilize Random Forest Classifier and Multinomial NB. The second method is deep learning, where we will use RNN and LSTM as well as N-gram CNN to analyze the dataset. Upon completing the model training, we will proceed to assess the model's performance through the evaluation process.

A. Machine Learning

The reason why we use this method is that texts containing sentiment usually have more complex patterns and semantics and are not structured data. Machine learning models can learn the sentiment expression features of these texts from a large number of texts, and generalize them to adapt to different sample distributions and sentiment expressions. At the same time, this approach can automatically process a large amount of new and unseen text data when the model has already been

Identify applicable funding agency here. If none, delete this.

trained. Compared with traditional rule-based or dictionary-based methods, machine learning approaches usually have higher accuracy and adaptability, and have stable performance in different text environments.

B. Deep Learning

There are some reasons for using deep learning methods for text sentiment analysis:

1. Strong Feature Learning and Representation
2. Contextual Understanding
3. Handling Long Texts
4. Better Performance
5. End-to-End Training

C. n-gram Convolutional Neural Networks (CNNs)

Using n-gram Convolutional Neural Networks (CNNs) for text sentiment analysis offers several advantages: 1. Feature Extraction

2. Multi-Scale Analysis
3. Contextual Understanding
4. Domain Adaptation
5. Computational Efficiency

To conclude, the utilization of n-gram Convolutional Neural Networks in text sentiment analysis enhances the model's capability to grasp the emotional context, extract features, and ultimately elevate the overall performance and precision of sentiment analysis.

IV. Experimental Setup

A. Dataset

Due to the large size of the dataset, this model only selected the first 20,000 data points for each emotion, totaling 40,000 data points. This approach also ensured a balanced sample size. We separated all the data by 8:2, 80% for training and 20% for validation. 1 means positive and 0 means negative. We also use word-cloud visuals to represent the words that appear frequently in the dataset labeled as positive and negative, respectively.

Fig. 1. Positive and Negative word-cloud

B. Machine Learning

The first step is EDA processing, which involves preprocessing the text. We use the regex method [10] to remove URL links and emojis from the textual content. And then, after processing, we eliminate missing values and duplicate values from the dataset.

1. remove outliers We need to remove outliers from the specified column ('text_lens') in the 'train' dataset.

2. tokenize and vocabularyize: We perform text preprocessing on the 'Tweet_content' column in the DataFrame 'train', and store the preprocessed results in a new column called 'preprocessed_text'.

The second step is transforming Original Text Data into TF-IDF Feature Matrix.

The third step is training the model. Our group chose the Random Forest and MultinomialNB methods to train the model. The Random Forest method is more stable while the MultinomialNB method is more efficient.

C. Deep learning

The first step is EDA processing, which entails preprocessing the text by removing sentiment-irrelevant words such as stop words, URLs, punctuation, repeated words, email addresses, and numbers.

The second step is preparing the training input features. Our group transforms the text words into an array format. A maximum of 500 features are chosen for training.

The third step is training the model. Implementing Tensorflow based model for training.

For the two-class scenario, we use "binary_crossentropy," while for situations involving more than two classes, we utilize "categorical_crossentropy."

An optimizer is a function used to adjust neural network characteristics, such as the learning rate, in order to minimize losses.

Training and Validation with Parameter Tuning: We employ the training dataset and allocate 10% of it for validation purposes.

The following parameters are configured: - A batch size of 128 implies the processing of 128 tweets during each iteration for model training.

- With an epoch count set to 8, the model engages in training on the dataset for a cumulative span of 8 cycles.

The fourth step involves making predictions on the test data.

D. n-gram CNNs model

A multi-channel convolutional neural network for document classification employs diverse adaptations of the standard model, each with distinct-sized kernels. This approach enables the document to be analyzed across various n-grams concurrently, enabling the model to adeptly assimilate and synthesize these different interpretations.

Based on the results of data preprocessing, we can know that the max document length is 29 and the vocabulary size is 46498.

To begin, we use one-dimensional convolution. If the window size is 3, it corresponds to the yellow portion of the diagram, indicating the selection of three words each time. Following the convolution, there's a subsequent step of max-1-pooling. This process entails selecting the most salient words or phrases from the sentence, which become the output of the subsequent layer. These identified keywords are then amalgamated and input into a fully connected layer to produce the final classification outcome. Constructing our multi-input model involves the utilization of the Keras functional API. The model comprises three input channels designed to process 4-grams, 6-grams, and 8-grams derived from the Twitter text.

The Conv1D layer, comprising 32 filters, is equipped with a kernel size aligned with the number of words under analysis during each iteration. A MaxPooling1D layer is employed for the amalgamation of the convolution layer's output.

V. Results

A. Machine learning

	precision	recall	f1-score	support
0	0.72	0.72	0.72	4000
1	0.72	0.73	0.72	4000
accuracy			0.72	8000
macro avg	0.72	0.72	0.72	8000
weighted avg	0.72	0.72	0.72	8000

Fig. 2. Random Forest Classifier

	precision	recall	f1-score	support
0	0.74	0.68	0.71	4000
1	0.71	0.77	0.74	4000
accuracy			0.72	8000
macro avg	0.73	0.72	0.72	8000
weighted avg	0.73	0.72	0.72	8000

Fig. 3. MultinomialNB

The biggest advantage of MultinomialNB method over the Random Forest method is the fast running speed. Based on the results of the two methods, it can be seen that the two methods agree on the final accuracy. The Random Forest method is more stable while the MultinomialNB method is more efficient.

B. Deep Learning

This is the result of training data. We can see that the accuracy of the deep learning method is 0.7416 and the loss is 0.5392.

This is the result of test data. We can see that the accuracy of the deep learning method is 0.7409 and the loss is 0.5295.

```
Epoch 1/8
225/225 [==============================] - 133s 584ms/step - loss: 0.6235 - accuracy: 0.6428 - val_loss: 0.5349 - val_accuracy: 0.7322
Epoch 2/8
225/225 [==============================] - 130s 577ms/step - loss: 0.5148 - accuracy: 0.7508 - val_loss: 0.5719 - val_accuracy: 0.6988
Epoch 3/8
225/225 [==============================] - 134s 594ms/step - loss: 0.4967 - accuracy: 0.7617 - val_loss: 0.5150 - val_accuracy: 0.7466
Epoch 4/8
225/225 [==============================] - 129s 574ms/step - loss: 0.4865 - accuracy: 0.7693 - val_loss: 0.5164 - val_accuracy: 0.7437
Epoch 5/8
225/225 [==============================] - 132s 586ms/step - loss: 0.4919 - accuracy: 0.7662 - val_loss: 0.5175 - val_accuracy: 0.7422
Epoch 6/8
225/225 [==============================] - 131s 580ms/step - loss: 0.4735 - accuracy: 0.7775 - val_loss: 0.5210 - val_accuracy: 0.7453
Epoch 7/8
225/225 [==============================] - 76s 337ms/step - loss: 0.4868 - accuracy: 0.7635 - val_loss: 0.5462 - val_accuracy: 0.7322
Epoch 8/8
225/225 [==============================] - 68s 300ms/step - loss: 0.4879 - accuracy: 0.7688 - val_loss: 0.5392 - val_accuracy: 0.7416
```

Fig. 4. result of training data

```
250/250 [==============================] - 8s 32ms/step - loss: 0.5295 - accuracy: 0.7409
250/250 [==============================] - 8s 33ms/step
```

Fig. 5. result of test data

We use the Confusion matrix and ROC CURVE to show our results. The following are the assessment metrics used to gauge the model's performance.

Correct predictions made by the trained model are represented by dark blue boxes, while sky blue boxes indicate incorrect predictions. A total of 2804 tweets were accurately predicted as negative sentiments. However, there were 879 tweets that were predicted as positive sentiments by the model but were actually negative sentiments. Furthermore, 3132 tweets were correctly predicted as positive sentiments. Conversely, there were 1194 tweets that were predicted as negative sentiments by the model but were, in reality, positive sentiments.

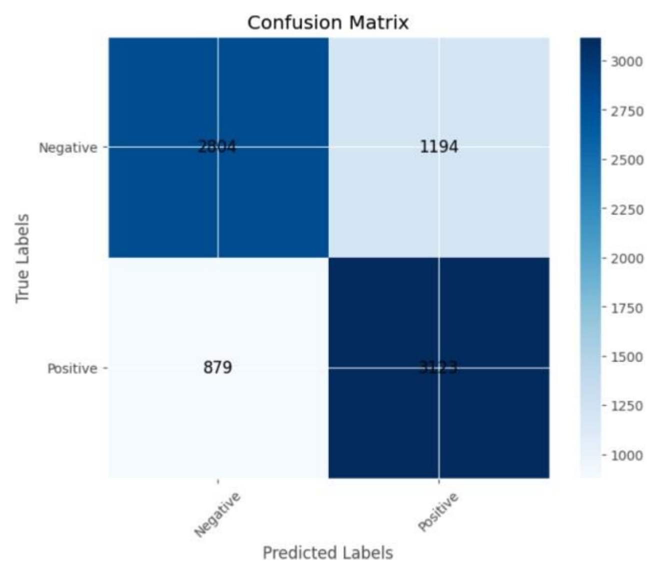

Fig. 6. confusion matrix

The ROC curve can be utilized for threshold selection and model comparison. The area under the ROC curve, known as AUC, can also be used for model comparison. A high AUC and a ROC curve close to the upper-left corner are indicators of good performance.

From epoch 1 to the last one, the trends of both are roughly similar, indicating that there is no overfitting or underfitting.

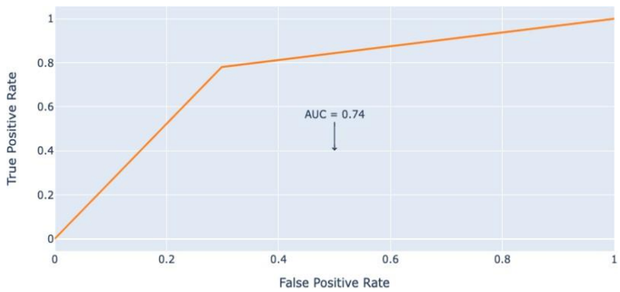

Fig. 7. False Positive Rate

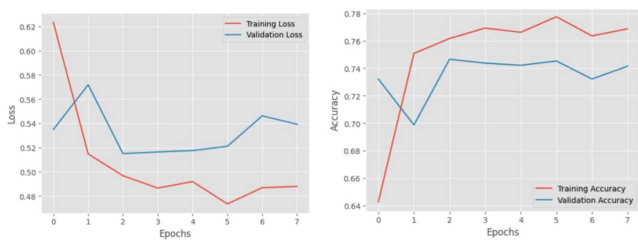

Fig. 8. Epochs

C. n-gram CNNs model

```
Epoch 1/7
313/313 [==============================] - 13s 41ms/step - loss: 0.5749 - accuracy: 0.6859 - val_loss: 0.4185 - val_a
ccuracy: 0.8571
Epoch 2/7
313/313 [==============================] - 13s 42ms/step - loss: 0.3502 - accuracy: 0.8533 - val_loss: 0.3999 - val_a
ccuracy: 0.8214
Epoch 3/7
313/313 [==============================] - 13s 42ms/step - loss: 0.1501 - accuracy: 0.9449 - val_loss: 0.3961 - val_a
ccuracy: 0.8214
Epoch 4/7
313/313 [==============================] - 13s 42ms/step - loss: 0.0709 - accuracy: 0.9756 - val_loss: 0.3438 - val_a
ccuracy: 0.8214
Epoch 5/7
313/313 [==============================] - 13s 41ms/step - loss: 0.0456 - accuracy: 0.9844 - val_loss: 0.3013 - val_a
ccuracy: 0.8571
Epoch 6/7
313/313 [==============================] - 13s 41ms/step - loss: 0.0322 - accuracy: 0.9894 - val_loss: 0.3324 - val_a
ccuracy: 0.8214
Epoch 7/7
313/313 [==============================] - 13s 41ms/step - loss: 0.0252 - accuracy: 0.9915 - val_loss: 0.2784 - val_a
ccuracy: 0.8929
```

Fig. 9. result

According to the output results of the model, the training accuracy reaches 0.9915, and the verification accuracy is close to 0.90.

From epoch 1 to the last one, the trends of both are roughly similar, indicating that there is no overfitting or underfitting.

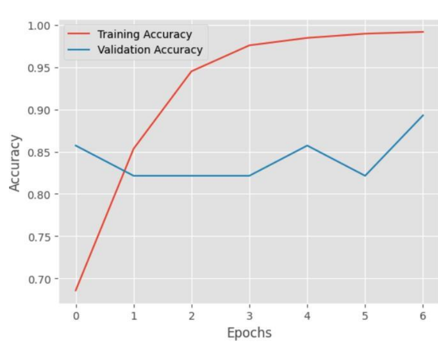

Fig. 10. Epochs

CONCLUSIONS

We choose Twitter as our target of sentiment analysis, because Twitter contains a huge number of emotions, and there are industrial interests to these emotions for their commercial uses. To implement this function, we have used different NLP techniques, both in Machine Learning and Deep Learning. For each method, we used different models and compared their accuracy rates. According to the results, we could see that the n-gram CNN model performed best among all models, which is close to 90%. On the other hand, RNN and LSTM model and ML method had worse performance. By analyzing the results, we could see some improvements in our experiment. Firstly, due to the word limit, which is 140 characters, a single tweet contains less information about the sequence of words, which would limit the performance of RNN models. Secondly, when doing the word pre-processing, one process is to remove all stop words, such as a, an, and other words that contain no sentiment information. However, there is a probability that some important words such as NO, could be removed, and hence could change the original sentiment information. To improve this, we could find out a better method to remove the stop words without affecting those important words. Finally, there are more different models that would have better performance on sentiment analysis. Among all of them, only 4 models shown above have been selected and implemented. We could implement more models for our further studies.

REFERENCES

[1] Wikipedia Contributors. “Exploratory Data Analysis.” Wikipedia, Wikimedia Foundation, 16 Mar. 2019, en.wikipedia.org/wiki/Exploratory_data_analysis.

[2] IBM. “What Is Exploratory Data Analysis? — IBM.” Www.ibm.com, 2020, www.ibm.com/topics/exploratory-data-analysis.

[3] [3]Heath, Nick. “What Is Machine Learning? Everything You Need to Know.” ZDNet, ZDNet, 14 Sept. 2018, www.zdnet.com/article/what-is-machine-learning-everything-you-need-to-know/.

[4] BRUSH, STEPHEN G. “History of the Lenz-Ising Model.” Reviews of Modern Physics, vol. 39, no. 4, 1 Oct. 1967, pp. 883–893, https://doi.org/10.1103/revmodphys.39.883.

[5] Amari, Shun-ichi, and Kenjiro Maginu. “Statistical Neurodynamics of Associative Memory.” Neural Networks, vol. 1, no. 1, 1 Jan. 1988, pp. 63–73, https://doi.org/10.1016/0893-6080(88)90022-6. Accessed 25 Apr. 2023.

[6] Hopfield, J. J. “Neural Networks and Physical Systems with Emergent Collective Computational Abilities.” Proceedings of the National Academy of Sciences, vol. 79, no. 8, 1 Apr. 1982, pp. 2554–2558, www.pnas.org/content/79/8/2554, https://doi.org/10.1073/pnas.79.8.2554.

[7] Soffer, Shelly, et al. “Convolutional Neural Networks for Radiologic Images: A Radiologist’s Guide.” Radiology, vol. 290, no. 3, Mar. 2019, pp. 590–606, https://doi.org/10.1148/radiol.2018180547. Accessed 22 Nov. 2019.

[8] W3Schools. “Python RegEx.” W3schools.com, 2019, www.w3schools.com/python/python_regex.asp.