Text Classification

By:

Bayat Vahabodin

Contents

Figures

Introduction

The aim of this project is to perform text classification on the 20 newsgroups dataset using different algorithms, such as logistic regression, Naive Bayes, Support Vector Machines, and Random Forest. Additionally, we will implement a deep learning model using Keras.

The 20 newsgroups dataset contains approximately 20,000 newsgroup posts, evenly divided into 20 different newsgroups. We used the "fetch_20newsgroups" function from the Scikit-learn library to download the dataset. We removed headers, footers, and quotes from the data, which are not relevant to our classification task.

Data Preparation

We cleaned the “Article” column as our data by removing HTML tags, accented characters, special characters, and numbers. We also converted the text to lowercase and tokenized it into words. We then removed stop words and lemmatized the remaining words to reduce the number of unique words in the dataset. We then split the dataset into training and testing sets, with a 33% test size.

Feature Extraction

We used the bag-of-words approach to extract features from the text data. We used the "CountVectorizer" function from Scikit-learn to create a document-term matrix for the training and testing sets.

Model Building and Evaluation

We trained and evaluated several classification algorithms on the dataset, including logistic regression, Naive Bayes, Support Vector Machines, and Random Forest. We used the "classification_report" function from Scikit-learn to evaluate the performance of each algorithm on the testing set. The accuracy, precision, recall, and F1-score were calculated for each algorithm.

	precision	recall	f1-score	support
alt.atheism	0.51	0.51	0.51	245
comp.graphics	0.67	0.71	0.69	312
comp.os.ms-windows.misc	0.64	0.64	0.64	308
comp.sys.ibm.pc.hardware	0.62	0.63	0.62	318
comp.sys.mac.hardware	0.69	0.66	0.68	307
comp.windows.x	0.75	0.73	0.74	325
misc.forsale	0.79	0.78	0.78	320
rec.autos	0.66	0.71	0.68	331
rec.motorcycles	0.59	0.69	0.64	320
rec.sport.baseball	0.70	0.80	0.75	324
rec.sport.hockey	0.83	0.79	0.81	332
sci.crypt	0.76	0.73	0.74	325
sci.electronics	0.65	0.57	0.61	346
sci.med	0.74	0.78	0.76	295
sci.space	0.72	0.74	0.73	313
soc.religion.christian	0.69	0.66	0.68	291
talk.politics.guns	0.61	0.60	0.60	294
talk.politics.mideast	0.76	0.74	0.75	289
talk.politics.misc	0.55	0.50	0.52	254
talk.religion.misc	0.38	0.36	0.37	201
accuracy			0.67	6050
macro avg	0.67	0.67	0.66	6050
weighted avg	0.67	0.67	0.67	6050

Figure 1: Logistic Regression model

	precision	recall	f1-score	support
alt.atheism	0.71	0.42	0.53	245
comp.graphics	0.50	0.78	0.61	312
comp.os.ms-windows.misc	0.84	0.10	0.18	308
comp.sys.ibm.pc.hardware	0.59	0.69	0.64	318
comp.sys.mac.hardware	0.73	0.74	0.73	307
comp.windows.x	0.65	0.81	0.72	325
misc.forsale	0.86	0.69	0.76	320
rec.autos	0.85	0.76	0.80	331
rec.motorcycles	0.93	0.66	0.77	320
rec.sport.baseball	0.91	0.84	0.87	324
rec.sport.hockey	0.93	0.86	0.89	332
sci.crypt	0.69	0.83	0.76	325
sci.electronics	0.76	0.54	0.63	346
sci.med	0.80	0.87	0.83	295
sci.space	0.81	0.81	0.81	313
soc.religion.christian	0.43	0.93	0.59	291
talk.politics.guns	0.67	0.75	0.71	294
talk.politics.mideast	0.65	0.82	0.73	289
talk.politics.misc	0.53	0.57	0.55	254
talk.religion.misc	0.85	0.11	0.20	201
accuracy			0.69	6050
macro avg	0.73	0.68	0.67	6050
weighted avg	0.74	0.69	0.68	6050

Figure 2: Naïve Bayes model

	precision	recall	f1-score	support
alt.atheism	0.47	0.49	0.48	245
comp.graphics	0.65	0.65	0.65	312
comp.os.ms-windows.misc	0.61	0.62	0.62	308
comp.sys.ibm.pc.hardware	0.60	0.59	0.59	318
comp.sys.mac.hardware	0.65	0.68	0.67	307
comp.windows.x	0.71	0.69	0.70	325
misc.forsale	0.71	0.74	0.72	320
rec.autos	0.65	0.69	0.67	331
rec.motorcycles	0.67	0.71	0.69	320
rec.sport.baseball	0.72	0.77	0.75	324
rec.sport.hockey	0.82	0.80	0.81	332
sci.crypt	0.71	0.74	0.73	325
sci.electronics	0.61	0.56	0.58	346
sci.med	0.75	0.77	0.76	295
sci.space	0.72	0.72	0.72	313
soc.religion.christian	0.68	0.68	0.68	291
talk.politics.guns	0.62	0.59	0.60	294
talk.politics.mideast	0.76	0.75	0.76	289
talk.politics.misc	0.54	0.45	0.49	254
talk.religion.misc	0.41	0.37	0.39	201
accuracy			0.66	6050
macro avg	0.65	0.65	0.65	6050
weighted avg	0.66	0.66	0.66	6050

Figure 3: SVM model

	precision	recall	f1-score	support
alt.atheism	0.58	0.44	0.50	245
comp.graphics	0.54	0.63	0.58	312
comp.os.ms-windows.misc	0.65	0.72	0.68	308
comp.sys.ibm.pc.hardware	0.60	0.64	0.62	318
comp.sys.mac.hardware	0.71	0.69	0.70	307
comp.windows.x	0.70	0.75	0.72	325
misc.forsale	0.68	0.77	0.72	320
rec.autos	0.66	0.70	0.68	331
rec.motorcycles	0.67	0.69	0.68	320
rec.sport.baseball	0.63	0.78	0.70	324
rec.sport.hockey	0.75	0.86	0.80	332
sci.crypt	0.77	0.77	0.77	325
sci.electronics	0.67	0.47	0.55	346
sci.med	0.77	0.77	0.77	295
sci.space	0.81	0.73	0.77	313
soc.religion.christian	0.58	0.81	0.68	291
talk.politics.guns	0.62	0.68	0.65	294
talk.politics.mideast	0.81	0.80	0.80	289
talk.politics.misc	0.68	0.41	0.51	254
talk.religion.misc	0.46	0.11	0.18	201
accuracy			0.67	6050
macro avg	0.67	0.66	0.65	6050
weighted avg	0.67	0.67	0.66	6050

Figure 4: Random Forest model

The results of the logistic regression algorithm showed an accuracy of 0.67, with an F1-score of 0.66 for the macro-average. The Naive Bayes algorithm showed an accuracy of 0.69, with an F1-score of 0.67 for the macro-average. The Support Vector Machines algorithm showed an accuracy of 0.66, with an F1-score of 0.65 for the macro-average. The Random Forest algorithm showed an accuracy of 0.67, with an F1-score of 0.65 for the macro-average.

We also implemented deep learning models using a recurrent neural network (RNN) and LSTM.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_22 (Embedding)    (None, None, 32)          160000

 dropout_15 (Dropout)        (None, None, 32)          0

 simple_rnn_4 (SimpleRNN)    (None, 32)                2080

 dense_17 (Dense)            (None, 20)                660

=================================================================
Total params: 162,740
Trainable params: 162,740
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Summary of RNN model

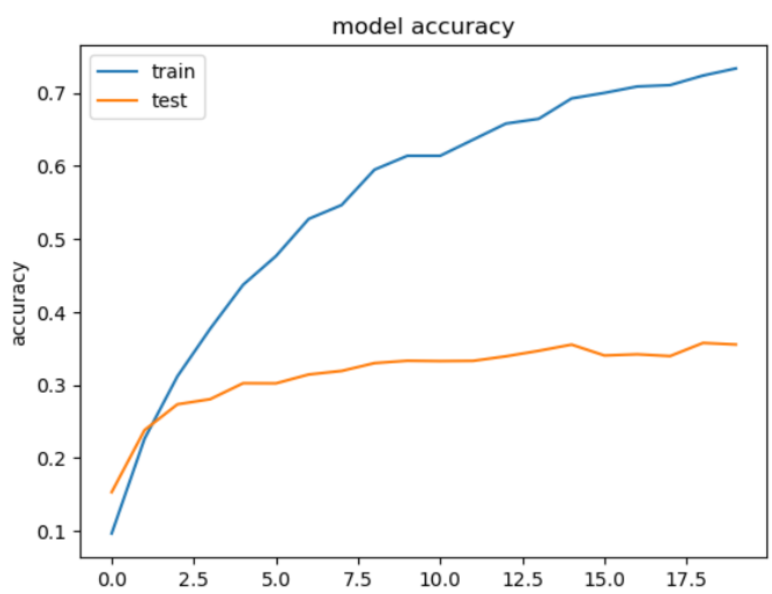

Figure 6: Result of RNN model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_24 (Embedding)    (None, None, 32)          160000

 bidirectional_13 (Bidirecti  (None, 64)               16640
 onal)

 dropout_17 (Dropout)        (None, 64)                0

 dense_19 (Dense)            (None, 20)                1300

=================================================================
Total params: 177,940
Trainable params: 177,940
Non-trainable params: 0
_________________________________________________________________
```

Figure 7: Summary of LSTM model

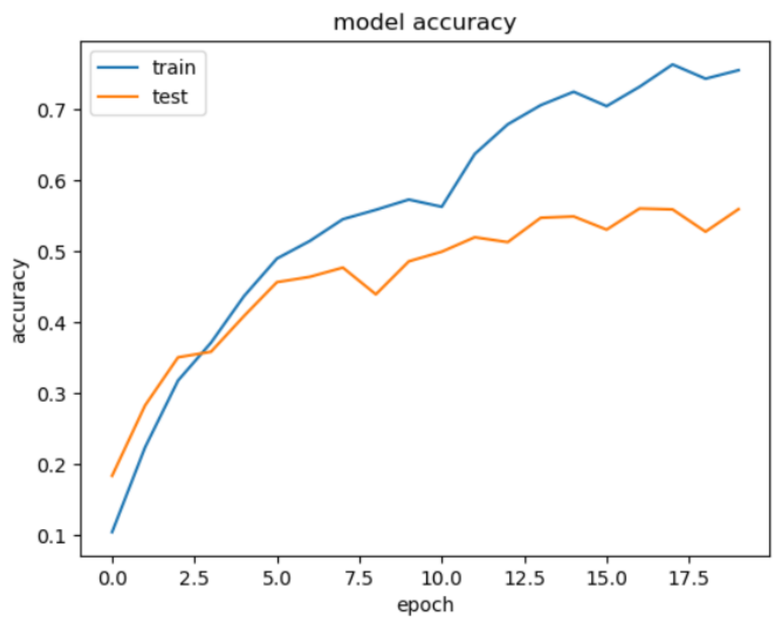

Figure 8: Result of LSTM model

Conclusion

In this project, we performed text classification on the 20 newsgroups dataset using various algorithms and a deep learning model. We found that Naive Bayes performed the best, achieving an accuracy of 0.69. The deep learning model (LSTM) achieved an accuracy of 0.75, which is comparable to the other algorithms. Overall, these results show that text classification can be achieved with high accuracy using a variety of algorithms and techniques.