News Classification Using Various Machine Learning Methods

Wei Yuan [1]

Abstract

This project focus on the news categories classification by using different machine learning algorithms, including traditional machine learning algorithms, such as: Logistic Regression, Naive Bayes, and Support Vector Machine with TD-IDF vectotizer. Also, the project dive into the deep learning algorithms to investigate the performance among FastText, LSTM, CNN. Then, the project compare among different machine learning algorithms to discuss their performance and disadvantages.

1. Introduction

Nowadays, many sources on the Internet generate a large amount of news every day. Also, the demand for information is continually growing, and it is crucial to enable users to access information of interest quickly and efficiently by classifying news.

In the text-mining, classification is a challenging area as it requires preemptive steps to convert unstructured data into structured information. As the number of news increases, it is difficult for a user to get the news that interests him, so it is necessary to classify the news to make it easily accessible. Categorization refers to grouping to make navigation between articles easier. Internet news needs to be divided into different categories. This will help the user access the news of their interest in real-time.

In this way, machine learning models for automatic news classification can be used to identify untracked news topics and present individual opinions. Untracked news and make individual suggestions based on the user's prior interests. Thus, our goal is to build models that take news headlines and short descriptions as input, output news categories and discuss their performance and various drawbacks among different models. And to build the model more efficiency, the project took the the top 10 categories as training data.

[1]Master of Science, Computer Science, York University, Toronto, Ontario, Canada.

Project Report for EECS 5327 (Winter 2021), Department of Electrical Engineering and Computer Science, York University.

2. Literature Reviews

2.1. GloVe

As the paper "Global vector representation of words" (Pennington et al., 2014) states, GloVe incorporates global statistics (word co-occurrence), and the Glove is a predictive Word2Vec model that predicts the context of the given word, while GLoVe learns by constructing a co-occurrence matrix that counts how often a word occurs in a context.

2.2. TextRNN for Text Classification

TextRNN by using Recurrent Neural Networks to solve the text classification problems by using LSTM. It proposed a multi-task learning framework to jointly learn multiple related tasks. Based on recurrent neural networks, and three different information sharing mechanisms to model the text with task-specific and shared layers (Liu et al., 2016).

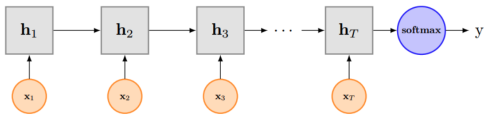

Figure 1. TextRNN for Text Classification

Long short-term memory (LSTM) networks are a special type of RNN capable of learning long-term dependencies (Hochreiter & Schmidhuber, 1997). work very well on a variety of problems and are now widely used. Recurrent neural networks capture contextual information by maintaining the state of all previous inputs. This model analyzes the text word by word and stores the semantics of all previous text in a fixed size hidden layer.

2.3. CNN for Sentence Classification

The Convolutional Neural Networks for Sentence Classification (Kim, 2014) proposed that when a particular pattern is detected, the result of each convolution is initiated. By changing the size of the kernels, connecting their outputs, you can make yourself detect patterns of multiple sizes. And the model uses multiple filters with different window sizes

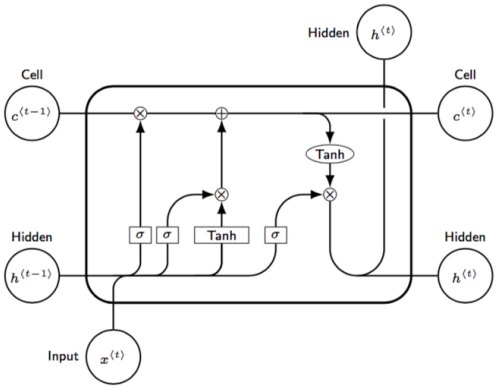

Figure 2. LSTM Network Structure

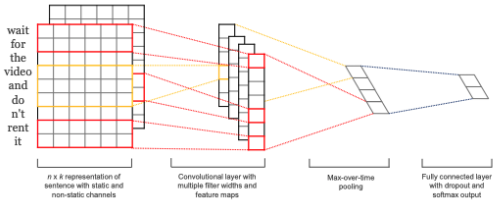

Figure 3. CNN for Sentence Classification Structure

to obtain multiple features.

2.4. RCNN for Text Classification

RCNN for text classfication model (Lai et al., 2015) applies a bidirectional recursive structure, which introduce considerably less noise compared to traditional window-based neural networks and maximize the capture of contextual information when learning word representations. In addition, the model can retain a larger range of word orderings when learning text representations. Next, the maximal set layer to automatically determine which features play a key role in text classification to capture the key components in the text. By combining the recursive structure and the maximum ensemble layer, the model takes advantage of both the recurrent neural model and the convolutional neural model.

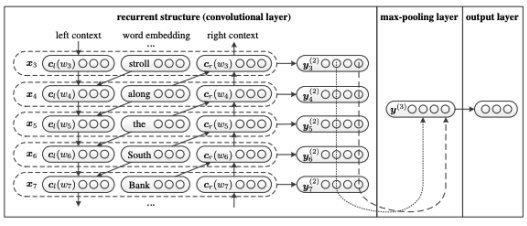

Figure 4. RCNN for Text Classification Structure

3. Methodology

3.1. News Data Collection

The project collect the dataset from Kaggle dataset that contain about 200853 rows of news data from 2012 to 2018 obtained from HuffPost, and based on the dataset, the news classification could be trained to identify the category for unstructued news article.

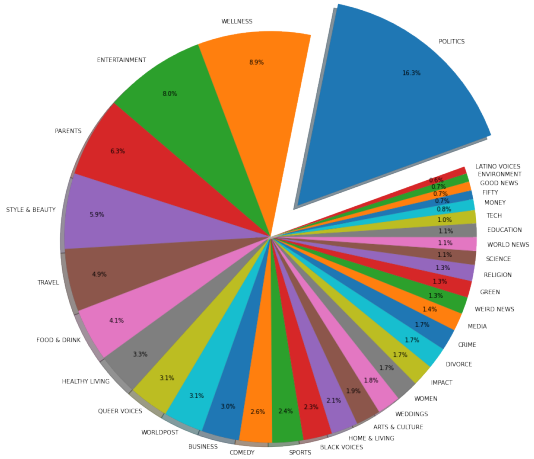

Figure 5. The Categories Distribution of News Dataset

Among the news dataset, there are 31 different categories, and each news consistently contains several attributes. To build a news categories classification, only "category", "title'" and "short description" attributes are helpful for the training model. Also, since the title and short description contain similar information, those two attributes can be merged into one attribute, "content", as input data of the classification model.

3.2. News Data Pre-processing

After the news texts are collected, the next step would be text pre-processing. Since this data comes from various data collection sources, it needs to be cleaned to free it from all the corrupt and useless data.

The data now needs to be distinguished from irrelevant words such as semicolons, commas, double quotes, periods, parentheses, special characters, etc. Also, the data has to be separated from those words that frequently appear in the text, called stop words.

After the tokenization and stopwords removal, the next important step would be to compare the performance between Bag of words (BOG) and TD-IDF (Term Frequency-Inverse Document Frequency) to do the input transformation. The purpose is to prepare for the input to be used for the following traditional machine learning models and GloVe as

WordEmbeeding method for the deep learning models.

3.2.1. TF-IDF

TF-IDF is an important search term importance measure in the field of information retrieval; it is used to measure the information that a keyword w can provide for a document.

Term Frequency (TF) indicates the frequency of occurrence of keyword w in document D_i.

$$TF_{w,D_i} = \frac{count(w)}{|D_i|}$$

where $count(w)$ be the occurrence of word w, and $|D_i|$ be the number of words in the document D_i.

Inverse Document Frequency (IDF) reflects the prevalence of a keyword, which means that the more prevalent a word is, the lower its IDF value; conversely, the higher the IDF value. the IDF is defined as follows:

$$IDF_w = \log \frac{N}{1 + \sum_{i=1}^{N} I\left(w, D_i\right)}$$

where N be the total number of documents, and $I(w, D_i)$ represents whether document D_i contains the keyword w.

Then, the value of TF-IDF of kwyword w is:

$$TF - IDF_{w,D_i} = TF_{w,D_i} * IDF_w$$

where N be the total number of documents, and $I(w, D_i)$ represents whether document D_i contains the keyword w.

3.2.2. WORD EMBEDDING

Word embedding is a form of word representation that connects human understanding of language with machine understanding. It has learned to represent text in an n-dimensional space in which words with the same meaning have similar representations. This means that two similar words are represented by almost similar vectors which are very close to each other in the vector space.

3.3. Evaluation Criteria

In this project, is uses several evaluation criteria to measure the model performance, including accuracy, F1 score, and AUC-ROC curve.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

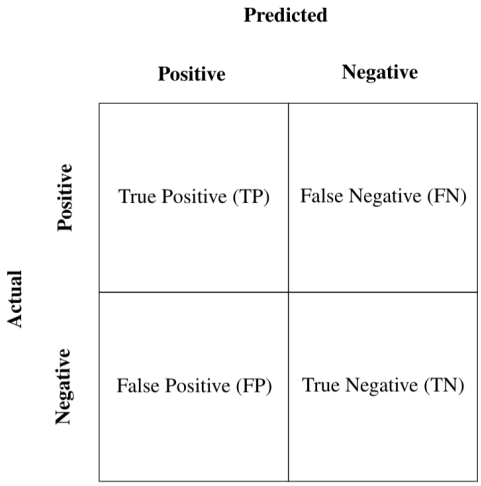

Figure 6. Confusion Matrix

3.3.1. ACCURACY

$$\text{Accuracy} = \frac{TP + TN}{N}$$

3.3.2. F1-SCORE

$$\text{F1} = \frac{2 * (Precision * recall)}{Precision + recall}$$

3.4. Traditional Machine Learning Algorithm

In the first part, the project uses the traditional machine learning algorithms to train the text categories classification model.

3.4.1. LOGISTIC REGRESSION

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a sigmoid function, and using the cross-entropy loss with L2 regularization to prevent the model over-fitting.

3.4.2. NAIVE BAYES

Naive Bayes is a probabilistic classifier based on text features. It computes probabilities and predicts labels for each instance. Naive Bayes does not consist of a single classification algorithm, but it includes many algorithms that train the classifier based on a single principle that states that the value of a particular feature is independent of the value of any other feature. The essence is that the value of a particular feature is independent of any other feature specified in a category. In the past classification, Naive Bayes was used in the classification of news articles. However, the

revised accuracy was reported due to its incorrect parameter evaluation. The best thing about the Naive Bayes algorithm is that it works equally well on both textual, digital, and numeric data, and it is easy to implement and compute.

3.4.3. Support Vector Machine

Support vector machine (SVM) is a supervised learning algorithm for fast and reliable classification that performs very well with a limited amount of data. However, if the size of the text file is large, then there will be many dimensions in the hyperspace, which may increase the computational cost of the process.

3.4.4. Random Forest

Random forest algorithm is a supervised classification algorithm that uses several trees to create a forest for the training model. It is also one of popular algorithms because it is simple and can be used for classification and regression problems. Random forests have almost the same hyperparameters as decision trees, it builds multiple decision trees and merges them together to obtain more accurate and stable predictions.

3.5. Deep Learning Neural Network Algorithms

In the second part of project, by unitizing the advanced deep learning neural network algorithms with word embedding features to train several neural network models in Long Short- Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), RCNN for classification and Self-Attention.

4. Experiments

After data collection, data pre-processing, then the project compare the performance among different machine learning models.

4.1. Traditional Machine Learning Models

The news category classifier with traditional machine learning algorithms follows the following process as figure 7 shown, and it consists of data collection, pre-processing, feature extraction with bag-of-word, and TF-IDF, then adapting the different traditional machine learning to train the news classifiers and evaluate it on the test dataset.

4.1.1. Logistic Regression

By applying the different methods of words represents, the result of logistic regression model shows as above. It states that the text classifier with bag of words achieved the higher F1-score about 0.62, but TF-IDF results the better performance at accuracy and AUC-ROC evaluation criteria at 0.64

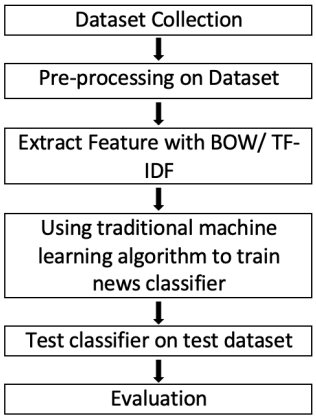

Figure 7. Traditional Machine Learning Procedure

and 0.95 respectively. Which provided the baseline for our project, and the AUC-ROC curve of logistic model with TD-IDF input transformation as following.

Methods	Accuracy	F1-Score	AUC-ROC
Bag-of Words	62.75	**62.08**	0.938
TF-IDF	**64.08**	61.99	**0.950**

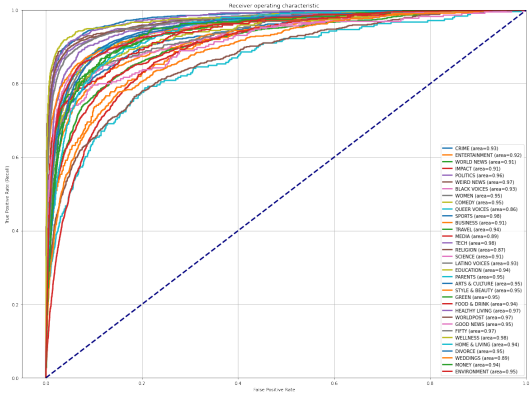

Figure 8. AUC-ROC curve of Logistic Regression

4.1.2. Naive Bayes

Similarly, the project used the multinormal Naive Bays classifier to train the two different input transformation methods.

Methods	Accuracy	F1-Score	AUC-ROC
Bag-of Words	**61.47**	57.49	0.9239
TF-IDF	61.00	57.11	**0.9517**

Among the bag of words and TF-IDF, the performance of bag of words on the multinormal Naive Bayes classifier

achieved the higher accuracy and F1-score about 0.6147 and 0.5749.

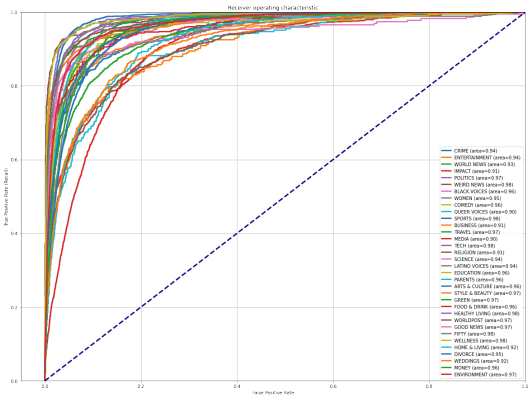

Figure 9. AUC-ROC curve of Naive Bayes

4.1.3. Support Vector Machine

By applying the linear kernel support vector machine into model, the model performance as shows below.

Methods	Accuracy	F1-Score	AUC-ROC
Bag-of Words	60.46	57.98	0.9216
TF-IDF	**65.10**	**63.52**	**0.9425**

With the training of the support vector machine, the classifier performance of accuracy increased to 0.65, and the F1-score increased to about 0.63, and the AUC-ROC achieved to 0.9425 with the TF-IDF.

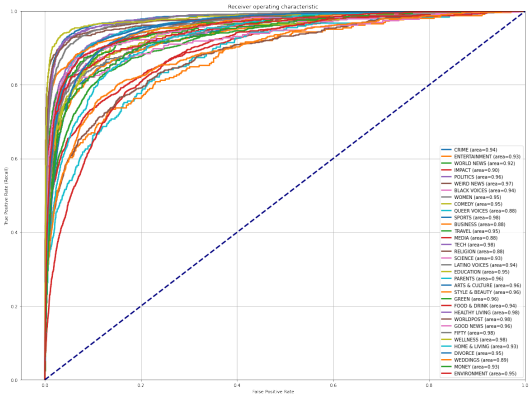

Figure 10. AUC-ROC curve of Support Vector Machine

4.1.4. Random Forest

However, after using the random forest into the news classifier, the results shows as following:

Methods	Accuracy	F1-Score	AUC-ROC
Bag-of Words	16.28	4.56	**0.7619**
TF-IDF	16.28	4.56	0.7570

It's clearly shows that the random forest algorithm dropped the model performance dramatically compare with other traditional machine learning algorithms.

In detail, the bag of words achieved a better model performance with random forest compare with that in TD-IDF feature extraction method.

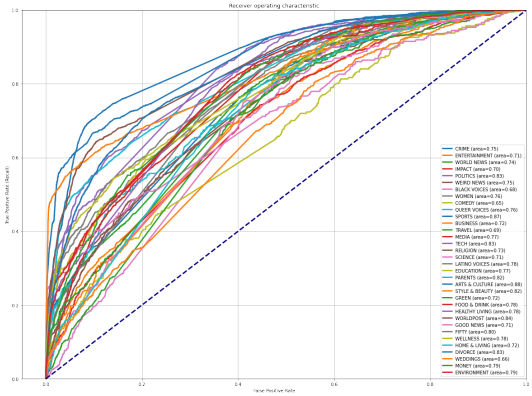

Figure 11. AUC-ROC curve of Random Forest

4.2. Deep Learning Algorithms

In the second part of the project, it introduced some different popular deep learning algorithm for the text classification tasks and its model performance.

There is a slight difference in model training pipeline with deep learning algorithm than it with traditional machine learning algorithm.

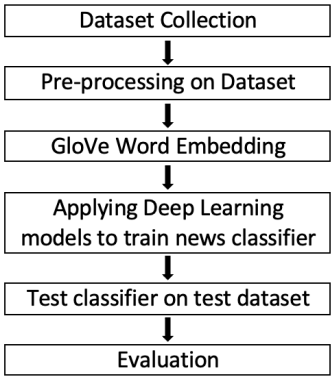

Figure 12. Deep Learning Algorithms Pipeline

After the data collection and pre-preocessing the dataset,

before training into the specific deep learning models, by using the GloVe (Pennington et al., 2014) word embedding method to improve the model performance and prevent overfitting. After word embedding, then using the different deep learning algorithms, LSTM (Hochreiter & Schmidhuber, 1997), BiLSTM, CNN for sentence classification (Kim, 2014), and textRCNN(Liu et al., 2016) to train the news category classification task.

4.2.1. LSTM

With the Long Short-Term Memory, the model could avoid the long-term dependency and remember information for long perio(Hochreiter & Schmidhuber, 1997).

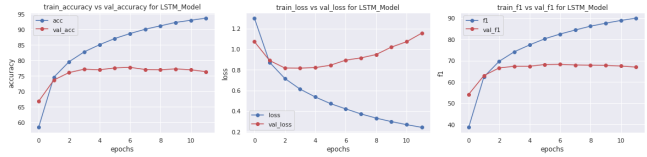

Figure 13. LSTM model performance

From the training process, it shows that the LSTM model achieved the best performance when the 7-th epoch, and the accuracy is 77.2 and the f1-score achieved to about 67.7

4.2.2. BiLSTM

With bidirectional LSTM, it will goes throught the content forward and reversely, and the model will correctly memory more content information, and the BiLSTM will improve the model performance.

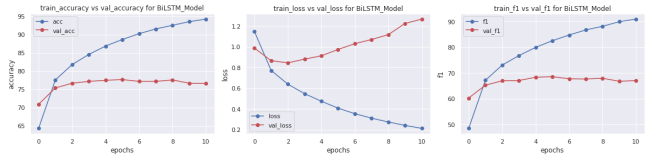

Figure 14. BiLSTM model performance

By improving the model setting of the original LSTM classifier to bidirectional LSTM, the model accuracy and F1-score are boost to 77.7 and 68.6 respectively.

4.2.3. CNN FOR SENTENCE CLASSIFICATION

The CNNs (Kim, 2014) by learning important words or phrases through selection by a max pooling layer, its model performance on the natural language processing for text classification problems as following.

The above training process shows that the CNN for text classification model has the best performance on accuracy and f1-score at the 7-th epoch, with the accuracy about 77.3 and f1-score at 68.3.

Figure 15. CNN for Sentence Classification Model Performance

4.2.4. TEXTRCNN

By combining both CNN and RNN, the deep learning algorithm of RCNN for text classification (Liu et al., 2016), firstly, it uses bidirectional RNN capture the contextual information. Next, by constructing a max-pooling layer to find the important features in text classification. In the end, the RCNN model will contains both advantage of RNN and CNN.

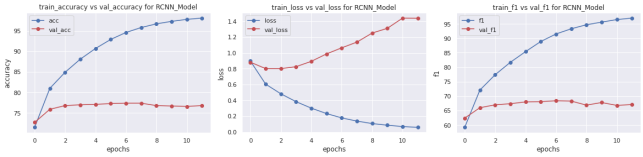

Figure 16. TextRCNN Model Performance

To improving the model setting of the RCNN by combine RNN and CNN, the model accuracy and F1-scoreare boost to 77.4 and 68.4 respectively.

5. Discussions

5.1. Comparison between different traditional machine learning algorithms

After implemented and trained the news category classifier, the experiments results shows as following:

Methods	Accuracy	F1-Score	AUC-ROC
LR + Bag of Words	62.75	62.08	0.938
LR + TF-IDF	64.08	61.99	0.950
NB + Bag of Words	61.47	57.49	0.9239
NB + TF-IDF	61.00	57.11	**0.9517**
SVM + Bag of Words	60.46	57.98	0.9216
SVM + TF-IDF	**65.10**	**63.52**	0.9425
RF + Bag of Words	16.28	04.56	0.7619
RF + TF-IDF	16.28	4.56	0.7570

Compare between different traditional machine learning algorithm on the news category classification task, from the above experiment results, it shows that the linear kernel support vector machine with TF-IDF algorithm achieved best performance on the test dataset with accuracy about 0.6510 and best F1-score of 0.6352. In addition, the Naive Bayes classifier with TF-IDF incresed the AUC-ROC score to 0.9517. Overall, the linear kernel support vector machine algorithm with TD-IDF improved the model performance on the news classification task.

Also, compare between the different feature extraction methods, bag of words and TD-IDF, and it states that the model with TD-IDF improved the accuracy, f1-score, and AUC-ROC score than the same model with bag of words feature extraction technique.

5.2. Comparison between Deep Learning Algorithms

After comparing the model performance between traditional machine learning models, it's necessary for the deep learning models with GloVe embedding (Pennington et al., 2014).

Methods	Accuracy	F1-Score
LSTM	77.2	67.7
BiLSTM	**77.7**	**68.6**
CNN	77.3	68.3
TextRCNN	77.4	68.4

Among the different deep learning algorithms, bidirectional LSTM model achieved the best performance on this project, news category classification task, with the accuracy of 77.7 and f1-score at 68.6.

6. Conclusion

The project focus on the news category classification task with different training models, especially in the traditional machine learning algorithms and advanced deep learning algorithms.

And the results shows that linear kernel support vector machine with TF-IDF feature extraction method performed the best performance with accuracy at 65.10 among the traditional machine learning algorithms. For the advanced deep learning algorithms, on this specific text classification task, the bidirectional LSTM, an advanced version of LSTM, achieved the highest accuracy score and f1-score with 77.7 and 78.6.

Generally, between traditional machine learning and advanced deep learning algorithms, the advanced deep learning algorithms have a huge (10%) improvement than traditional machine learning algorithms.

References

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

Kim, Y. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

Lai, S., Xu, L., Liu, K., and Zhao, J. Recurrent convolutional neural networks for text classification, 2015. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745/9552.

Liu, P., Qiu, X., and Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.