Sentiment Analysis on Movie Reviews

Ruoran Liu, *Western University*, Xueru Ye, *Western University*, and Keith Herbert, *Western University*

***Abstract*—Businesses often require Natural Language Processing (NLP) to derive business insights. This project analyses long-short term memory (LSTM), Distillated-Bidirectional Encoder Representations from Transformers (DistilBERT)-Logistic Regression Hybrid and Extreme Gradient Boosting (XGBoost) model's ability to predict fine-grained sentiment analysis of Rotten Tomato movie reviews. Model hyperparameters are tuned using Grid Search. Methods of analysis include a ranked matrix, hold out validation and K-Fold cross-validation. The results of analysis show that DistilBERT-Logistic Regression Hybrid is the most accurate model followed by XGBoost and LSTM models.**

INTRODUCTION

Natural language processing that derives meaning from language is critical to many businesses. Social media monitoring is often used to identify user sentiment regarding new products and services. Surveys are used to determine useful user insights and email providers use natural language processing to filter spam e-mails.

The project goal is to determine fine-grained sentiment analysis for the Rotten Tomatoes movie review dataset using the LSTM, DistilBERT-Logistic Regression Hybrid and XG Boost models. Hyper parameters are tuned using Grid Search to improve model accuracy. Sentiment categories to predict for movie reviews are negative, somewhat negative, neutral, somewhat positive and positive.

The background of models and optimizers used is covered in Section II, Background. Section III, Methodology, covers the model and data. After the implementation of all models, results will be compared using a ranked matrix for the final report and explained in Section IV, Evaluation. Section V, Conclusions, summarizes the results of the project.

BACKGROUND

Models

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides parallel tree boosting, also known as Gradient Boosting Decision Tree (GBDT) and Gradient Boosting Algorithm (GBM) that solves many data science problems in a fast and accurate way. XGBoost currently runs on major distributed environments such as Hadoop, SGE and MPI.

The DistilBERT-Logistic Regression model Hybrid consists of multiple models described below [1][2].

1) BERT is a bidirectional, unsupervised language representation model that has been pre-trained using Wikipedia. It uses surrounding text to determine the context of words and can read text in both directions at the same time.
2) The DistilBERT model is a faster version of BERT. It has 40% less parameters than BERT and runs 60% faster with 95% of the accuracy of BERT.
3) Logistic Regression is a model that is used to classify data. It uses input values with weights to predict an output value by transforming its output using a sigmoid function. The function returns a probability that can be used for classification.

LSTM is a Recurrent Neural Network (RNN) architecture which is designed to model sequential data and its long-range dependencies accurately. Unlike a feedforward Neural Network, the outputs of

RNN layers are fed as input to the current step.

LSTM architecture is composed of a cell state and a hidden state. The cell is the memory part of the LSTM unit, where the information from a previous interval is stored. There are three gates known as an input gate, an output gate and a forget gate in the hidden state. The cell makes decisions about what to store and modify in the gates. The purpose of adding these gates is to make the weights and biases in each layer adjustable and to link distant occurrences to a final output when solving a long sequence problem [3].

Related Work

Deep learning techniques have significantly out-performed traditional methods in several NLP tasks, and Deep Learning for Sentiment Analysis is one of the most popular research fields. In the past, most sentiment prediction systems work by looking at words in isolation, giving positive points for positive words and negative points for negative words and then summing up these points. Two of the most popular deep learning techniques for sentiment analysis are Convolutional Neural Networks (CNN)s and LSTMs. Even though CNNs are designed for computer vision, the fact that they are fast and easy to train, make them a popular choice for NLP problems. However, CNNs have no notion of order, thus when applying them to NLP tasks the crucial information of the word order is lost [4]. In this research, we are using models that compute the sentiment based on a sentence's meaning so the complex model will not only classify using single words.

Methodology

Frameworks

This project is created using the Python environment. Several packages from the Python ecosystem are used for our framework. NumPy, a package that provides multidimensional array objects for fast operations on arrays [5]. Torch, an open-source deep learning library that is used for loading datasets and the LSTM and DistilBERT models. Scikit-learn, a machine learning library that provides a wrapper interface for the XGBoost model [6]. Also, Scikit-learn that provides model selection and cross-validation methods for model evaluation [7].

Data

The dataset is comprised of tab-separated files with 156060 phrases from the Rotten Tomatoes dataset. There are 8544 sentences that have been shuffled from their original order. Each sentence has been parsed into many phrases by the Stanford parser. Each row consists of PhraseId, SentenceId, phrase content and its corresponding sentiment label. Phrases that are repeated such as short or common words are only included once in the data. Table I shows the structure of the data.

Table I. Data Structure

PhraseId	SentenceId	Phrase	Sentiment
1	1	This quiet , introspective and entertaining independent is worth seeking .	4
2	1	This quiet , introspective and entertaining independent	3
3	1	This	2
4	1	quiet , introspective and entertaining independent	4
5	1	quiet , introspective and entertaining	3

There are 5 sentiment labels in the dataset as shown in Table II.

Table II. Data Classes

Class	Description
0	Negative
1	Somewhat Negative
2	Neutral
3	Somewhat Positive
4	Positive

The distribution of the 5 sentiment labels in the dataset is displayed in Fig. 1.

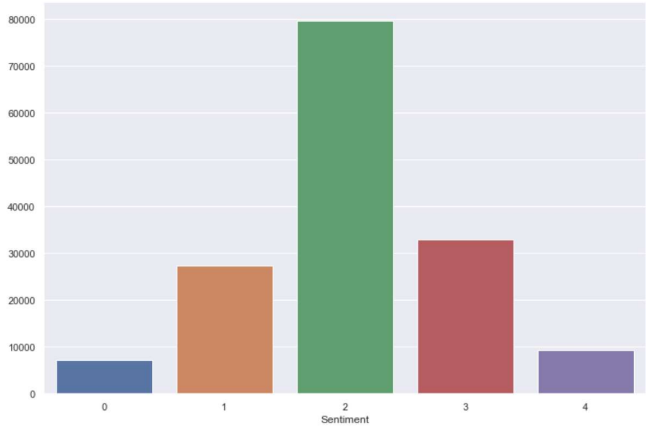

Fig. 1. Data Distribution

It can be inferred from the table above that the distribution of the data follows the Gaussian distribution, where neutral reviews phrases make up almost half of the dataset. For this project, we randomly select 2000 records from the dataset due to a limitation in computer processing power.

Data Preprocessing

Data preprocessing consists of the following steps:

1. Remove all punctuations in phrases
2. Remove all stop words such as the words "is" and "are" in phrases since they usually appear but are mostly irrelevant to a phrase's sentiment
3. Lemmatizing words so that different words express similar or the same meaning. These words can be considered as the same words. For example, people and person can be considered as the same word
4. Turn words into stems so that words that share the same origin and express the same sentiment can be considered the same

Models

The DistilBERT-Logistic Regression Hybrid model's data input text is transformed into numbers using a tokenizer. After tokenization, an array of numbers that represent sentences is created. The sentences are padded with zeros, so that each sentence have the same length. Next, DistilBERT is configured so that it ignores the padded values when processing the input. The model then determines sentence embeddings and these values are passed to the Logistic Regression model for processing. The Logistic Regression model is trained using DistilBERT output with labels. Next, Logistic Regression finds the relationships of the data to predict classes [8].

The LSTM model's data input text is transformed into numbers using a tokenizer. Each sentence is represented in the form of a list of numbers. The sentences are padded with zeros starting from the beginning of the sequence so that each sentence has the same length. The LSTM model computes a mapping from an input sequence $x = (x1, x2, \ldots x_n)$ to an output sequence $h = (h1, h2, \ldots h_n)$. In this case, the model takes a list of numbers as input to generate a classification as output.

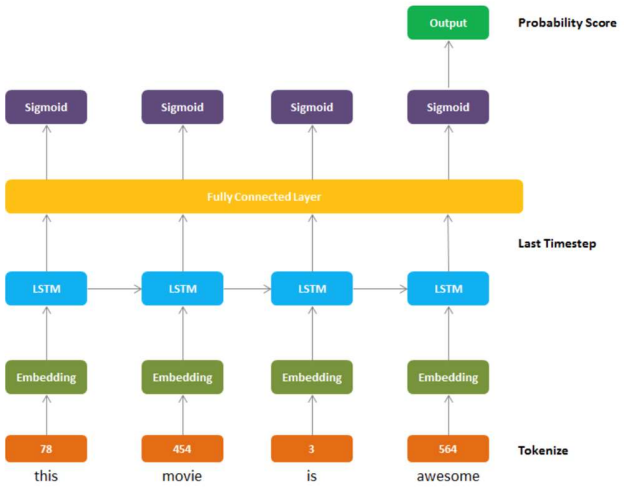

Fig. 2. LSTM Model Architecture

For each phrase in the review, the LSTM model uses the current number embedding and its previous hidden state to compute the next hidden state [9].

XGBoost is a scalable machine learning system for tree boosting. It implements the decision tree algorithm with gradient boosting technology. it uses decision trees as basic models and constructs decision trees in a stage-wise fashion like other boosting methods. Next, trees are generalized by allowing optimization of an arbitrary differentiable loss function. First the phrases are converted into vectors using TF-IDF algorithms. Next, vectors are put into the XGBoost model using the Sklearn XGBoost Classifier interface. The model will be

trained and tested [10].

Grid Search

An exhaustive search through a manually specified subset of the hyperparameter space of a learning algorithm. Grid Search tries all data subset combinations to determine the most accurate results.

The hyperparameters below were tuned.

XGBoost

1. Learning rate
2. Number of estimators
3. Max Depth

DistilBERT

1. Inverse of regularization strength

LSTM

1. Learning rate
2. Number of nodes in embedded layer(s)
3. Number of nodes in hidden layer(s)

Validation

Hold-out validation separates a shuffled dataset into training, validation and testing sets. The ratio of the training and testing sets for the XGBoost and DistilBERT-Logistic Hybrid models are 75% training and 25% testing as shown in Table III. Also, the dataset is split into training, validation and testing sets with the ratio of 80, 10 and 10 respectively for the LSTM model as shown in Table IV. Table V describes how the different data sets are used. In addition, Fig. 3 and Fig. 4 display a visual of how data is split where the blue and red areas represent the training data, the orange area represents the validation set and the green area represents the testing set.

Table III. XGBoost and DistilBERT Data Split

Training	75%
Testing	25%

Table IV. LSTM Data split

Training	80%
Validation	10%
Testing	10%

Table V. Dataset Usage

Purpose	Result	Model Use	Purpose Result Used for Model Training	Used for Tuning of Parameters
Training Set	Pattern Discovery	Accurate model prediction	Yes	Yes
Validation Set	Determine model behavior; Determine results of unseen data	Understand model tuning	No	Yes
Testing Set	Understand real world model performance on unseen data	Unbiased model estimate	No	No

Fig. 3. XGBoost and DistilBert-Logistic Regression Hybrid Holdout Data Visualization

Fig 4. LSTM Holdout Data Visualization

K-fold cross validation splits data into equal parts so that the holdout method can be performed on each part. Each iteration requires k-1 training sets and a k test set. The training process is complete after the testing of all data parts. 5-fold validation method is applied to all models. Fig. 5 displays the XGBoost and DistilBERT-Logistic Regression Hybrid model dataset visual where the red area represents the training data and the green represents a test set in which the model is iteratively tested. For the LSTM model, as shown in Fig. 6, The blue area represents the training data, the orange area represents the validation set, and the green area depicts a test set [11].

Fig. 5. XGBoost and DistilBert-Logistic Regression Hybrid K-fold Data Visualization

Fig. 6. LSTM K-fold Data Visualization

The classification report and confusion matrix are used to display classifier results. The classification report is used to display true positives, true negatives, false positives and false negatives of the classifier. The confusion matrix is a table that displays the actual and predicted classified values.

EVALUATION

Models

DistilBERT-Logistic Regression Hybrid model is used to predict movie review sentiment using K-Fold cross validation and hold out cross validation. The model is also refined using Grid Search optimization. Its initial hyperparameter consists of an inverse of regularization strength of 1. This produces a K-Fold accuracy of 64.9 percent with a standard deviation of 3.5 percent and a holdout accuracy of 70.8 percent.

Grid search optimization results in a regularization strength parameter of 0.3165. Optimised K-Fold cross validation produces an accuracy of 66.2 percent with a 4.3 percent standard deviation. Hold out validation produces slightly better results with an accuracy of 73 percent. For holdout validation, the confusion matrix shows that the algorithm classifies sentence sentiment accurately with some errors that are generally classified in adjacent classes. In addition, the classification report shows that the class with the most precision is positive followed by neutral, somewhat negative, somewhat positive and negative. Furthermore, the class with the highest recall is neutral followed by somewhat positive, somewhat negative, positive and negative. In addition, the class with the highest f1-score is neutral followed by positive, somewhat negative, somewhat positive and negative.

The prediction time of the model results in a training time of 217 seconds and a prediction time of 0.004 seconds. Storage memory consists of a training memory storage of 481.9 MiB and a prediction memory storage of 481.9 MiB. Fig. 7 provides a summary of the results.

DistilBERT-Logistic Regression Hybrid

Holdout Validation - Confusion Matrix

Real V/Predicted >	Class 0	Class 1	Class 2	Class 3	Class 4
Class 0	1	4	4	0	0
Class 1	2	34	45	3	0
Class 2	0	10	295	8	0
Class 3	0	3	41	30	0
Class 4	0	0	5	7	7

Holdout Validation - Classification Report

	Precision	Recall	f1-score	support
0	0.33	0.11	0.17	9
1	0.67	0.4	0.5	84
2	0.76	0.94	0.84	313
3	0.62	0.4	0.49	75
4	0.88	0.37	0.52	19
Accuracy			0.73	500
Macro Avg.	0.65	0.45	0.5	500
Weighted Avg.	0.72	0.73	0.71	500

KFold Validation - Accuracy	Mean 0.662000 (STD 0.042965)

Time and Memory Usage			
	Training	**Predicting**	
Time (seconds)	217.1079	0.004594	
Memory (MiB)	481.9	481.9 (no additional memory)	

Fig. 7. Optimised DistilBERT-Logistic Regression Hybrid Model Results

The LSTM model is also used to predict movie review sentiment using K-Fold cross validation and hold out cross validation. The model is refined using Grid Search optimization. Its initial hyperparameters consist of an embedded layer node value of 40, hidden layer node value of 25 and a learning rate of 0.04. This produces a K-Fold accuracy of 49 percent with a standard deviation of 0.5 percent and a holdout accuracy of 0.1 percent.

Grid search optimization results in an embedded layer node value of 80, hidden layer node value of 50 and a learning rate of 0.02. Optimised K-Fold cross validation produces an accuracy of 49.5 percent with a standard deviation of 0.5 percent. Hold out validation produces slightly better results with an

accuracy of 53 percent. For holdout validation, the confusion matrix shows that the algorithm classifies sentence sentiment poorly other than the neutral class. The classification report shows that the class with the most precision is neutral followed by somewhat negative, somewhat positive, negative and positive. Furthermore, the class with the highest recall is neutral followed by somewhat positive, somewhat negative, somewhat positive, negative and positive which all have an accuracy of zero. In addition, the class with the highest f1-score is neutral followed by somewhat negative, somewhat positive, negative and positive. We believe that the poor classification of classes other than neutral is due to the lack of training data. Further testing revealed that, when the dataset increased, class accuracy increased.

The prediction time of the model results in a training time of 107 seconds and a prediction time of 0.79 seconds. Storage memory consists of a training memory storage of 266.4 MiB and a prediction memory storage of 274.5 MiB. Fig. 8 provides a summary of results.

LSTM

Holdout Validation - Confusion Matrix

Real V/Predicted >	Class 0	Class 1	Class 2	Class 3	Class 4
Class 0	0	0	9	0	0
Class 1	0	0	37	0	0
Class 2	0	0	105	0	0
Class 3	0	0	39	0	0
Class 4	0	0	10	0	0

Holdout Validation - Classification Report

	Precision	Recall	f1-score	support
0	0	0	0	9
1	0	0	0	37
2	0.53	1	0.69	105
3	0	0	0	39
4	0	0	0	10
Accuracy			0.53	200
Macro Avg.	0.11	0.2	0.14	200
Weighted Avg.	0.28	0.53	0.36	200

KFold Validation - Accuracy Mean 0.495937 (STD 0.005165)

Time and Memory Usage

	Training	Predicting
Time (seconds)	107.0004	0.794834
Memory (MiB)	266.4	274.5

Fig. 8. Optimised LSTM Model Results

The XG Boost model is used to predict movie review sentiment using hold out validation. The model is also refined using Grid Search optimization. Its initial hyperparameters consist of the number of estimator's value of 100, max depth of 10, and learning rate of 0.1. This produces a K-Fold accuracy of 53.7 percent with a standard deviation of 3.8 percent and a holdout accuracy of 53 percent.

Grid search optimization results in an estimator's value of 8, max depth of 6 and learning rate of 0.51. Optimised K-Fold cross validation produces an accuracy of 58 percent. For holdout validation, the confusion matrix shows that the algorithm classifies sentence sentiment accurately with some errors that are generally classified in adjacent classes. Also, the classification report shows that the class with the most precision is negative followed by neutral, somewhat positive, negative and somewhat positive. Furthermore, the class with the highest recall is neutral followed by negative, positive, somewhat negative and somewhat positive. In addition, the class with the highest f1-score is neutral followed by somewhat positive, positive, negative and somewhat positive.

The prediction time of the model results in a training time of 3 seconds and a prediction time of 0.066 seconds. Storage memory consists of a training memory storage of 279.6 MiB and a prediction memory storage of 279.6 MiB. Fig. 9 provides a summary of results.

XGBoost

Holdout Validation - Confusion Matrix

Real V/Predicted >	Class 0	Class 1	Class 2	Class 3	Class 4
Class 0	3	3	23	2	1
Class 1	1	4	89	0	1
Class 2	0	0	258	1	1
Class 3	0	1	89	0	1
Class 4	1	0	20	0	1

Holdout Validation - Classification Report

	Precision	Recall	f1-score	support
0	0.6	0.09	0.16	32
1	0.5	0.04	0.08	95
2	0.54	0.99	0.7	260
3	0	0	0	91
4	0.2	0.05	0.07	22
Accuracy			0.530	500
Macro Avg.	0.370	0.230	0.200	500
Weighted Avg.	0.420	0.530	0.390	500

KFold Validation - Accuracy Mean 0.532

Time and Memory Usage		
	Training	**Predicting**
Time (seconds)	3.002289	0.066018
Memory (MiB)	279.6	279.6 (no additional memory)

Fig. 9. XGBoost Model Hold-out Validation

Ranked Matrix

The accuracy of the models is compared using a ranked matrix. The algorithms are evaluated using K-Fold and Holdout Validation and are equally weighted. The results of the matrix show that the DistilBERT-Logistic Regression Hybrid model is the most accurate model, from the ones tested, with a rank of 1 followed by the XGBoost and LSTM models as shown in Table VI.

Table VI. Ranked Matrix

	XGBoost	LSTM	DB Hybrid
K-Fold	2	3	1
Holdout	2	3	1
Total Score	4	6	2
Rank	2	3	1

We believe that the LSTM algorithm performs poorly because it does not sufficiently consider post word information because it reads sentences in the forward direction. XGBoost outperforms LSTM because it learns a better tree structure by using Hessian, a higher-order approximation. It makes trees deeper while keeping the variance low and is fast when compared to other implementations of gradient boosting [12]. However, it doesn't achieve the accuracy of the DistilBERT-Logistic Regression Hybrid model because it's sensitive to outliers since every classifier is obliged to fix the errors in the predecessors. Thus, this method is too dependent on outliers. Another problem is that the method is almost impossible to scale up since every estimator bases its accuracy on the previous predictors, making the procedure difficult to streamline [13]. The main reason why the DistilBERT-Logistic Regression Hybrid model performs so well in the movie review classification is because it applies bidirectional training of a transformer, a type of attention model. This allows the model to have a greater understanding of the context of words in a sentence.

CONCLUSIONS

Businesses often require natural language processing to derive business insights. This project analyses LSTM, DistilBERT-Logistic Regression Hybrid and XGBoost model's ability to predict fine-grained sentiment analysis of Rotten Tomato movie reviews. Methods of analysis include a holdout validation and K-Fold cross-validation.

Results of analysis show that the DistilBERT-Logistic Regression Hybrid is the most accurate model with a holdout accuracy of 73 percent followed by XGBoost with an accuracy of 53 percent and LSTM with an accuracy of 53 percent.

Future research involves an increase in dataset size, implementation of Bayesian Optimization, an increase in computer processing power and an analysis of other hybrid model combinations to improve model results.

REFERENCES

[1] J. K. "DistilBERT," DistilBERT - transformers 2.5.1 documentation. [Online]. Available: https://huggingface.co/transformers/model_doc/distilbert.html. [Accessed: 19-Mar-2020].

[2] "Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing," Google AI Blog, 02-Nov-2018. [Online]. Available: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html. [Accessed: 19-Mar-2020].

[3] "Long short-term memory " Wikipedia, 17-Mar-2020. [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accessed: 20-Mar-2020].

[4] “DataStories at SemEval-2017 Task 4: Deep LSTM with Attention forMessage-level and Topic-based Sentiment Analysis”, ACL Anthology, Aug-2017. [Online]. Available: https://www.aclweb.org/anthology/S17-2126.pdf. [Accessed: 20-Mar-2020].

[5] “What is NumPy?” What is NumPy? - NumPy v1.17 Manual. [Online]. Available: https://docs.scipy.org/doc/numpy/user/whatisnumpy.html. [Accessed: 19-Apr-2020].

[6] “Python API Reference” Python API Reference - xgboost 1.1.0-SNAPSHOT documentation. [Online]. Available: https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn. [Accessed: 19-Apr-2020].

[7] K. Jain, Kunal, and IIT Bombay in Aerospace Engineering, “Scikit-Learn In Python - Important Machine Learning Tool,” Analytics Vidhya, 01-Mar-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/. [Accessed: 19-Apr-2020].

[8] A. Pant, “Introduction to Logistic Regression,” Medium, 22-Jan-2019. [Online]. Available: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148. [Accessed: 19-Mar-2020].

[9] S. Agrawal, “Sentiment Analysis using LSTM” Medium, 18-Feb-2019. [Online]. Available: https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step-50d074f09948. [Accessed: 19-Mar-2020].

[10] T . Chen, C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Cornell University, 9-Mar-2016. [Online]. Available: https://arxiv.org/abs/1603.02754. [Accessed: 19-Mar-2020].

[11] Data Driven Fundamental Factor Modeling - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/A-visualization-of-K-fold-cross-validation-The-blue-area-represents-the-training-data_fig2_323561095 [accessed 19 Apr, 2020]

[12] A Gentle Introduction to XGBoost for Applied Machine Learning Jason Brownlee - https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine- [Accessed: 10-April-2020]

[13] Gradient Boosting - Overview, Tree Sizes, Regularization https://corporatefinanceinstitute.com/resources/knowledge/other/gradient-boosting [Accessed: 23-May-2020]