Predicting Scores and Controversialities of Reddit Posts Using Machine Learning

Fan Jue(A0221578B)[1], Han Jiyao(A0221330A)[2], Li Bozhao[3], Tian Xiao(A0220592L)[4], Xin Zhe(A0205928Y)[5]

National University of Singapore[1, 2, 3, 4, 5]

e0559110@u.nus.edu[1], e0556522@u.nus.edu[2], e0425559@u.nus.edu[3], e0555784@u.nus.edu[4], e0425851@u.nus.edu[5]

Abstract

Reddit posters may post contents that invoke disagreements and disputes, but in many cases they would not foresee these to happen. The extent of disagreements and disputes can be quantified by the score and controversiality data from the Reddit dataset. We aim to predict these two quantities based on the content of posts. In this report, we explored 9 machine learning models and 3 deep learning models and compared their performances. The best-performing models will be adopted as the driving core of our application – "PostLy", through which we are able to provide timely feedback and suggestions for posters about their drafts to help them make better decisions. We also include the main features and user interfaces of PostLy to demonstrate how it can be used together with Reddit.

Roles of Each Team Member

- Fan Jue: Attending meetings, exploring machine learning models, designing prototypes, drafting reports
- Han Jiyao: Attending meetings, exploring deep learning models, running data, drafting reports
- Li Bozhao: Attending meetings
- Tian Xiao: Attending meetings, exploring deep learning models, running data, analyzing results, drafting reports
- Xin Zhe: Attending meetings, preprocessing data, exploring machine learning models, drafting reports

Introduction

In the era of information explosion, Reddit has become one of the largest social media platforms where netizens share, disseminate information and engage in conversations with others on various topics. However, the anonymous nature of virtual discussion also weakens users' sense of responsibility for the content they post online. There has also been an emerging trend of online "toxicity" in the form of hate speech and online harassment (Mohan et. al. 2017). Especially during the outbreak of the Covid-19 pandemic, the widespread health misinformation and hate speech have led to an increase in anxiety and fear, as suggested by Medical News Today. Influential posts containing controversial topics such as "Wuhan virus" have also proven their capabilities in triggering disputes and causing discrimination against certain groups of people. Hence, in order to encourage the creation of less misleading and less provocative content on Reddit, we hope to allow frequent posters to be aware of the potential impact and the controversiality of their posts. Our project aims to help users make more informed decisions before disseminating information to the public and hence help to maintain an environment conducive to discussion on Reddit. Therefore, we will be exploring the possibility of using 9 machine learning techniques and 3 deep learning techniques to predict the acceptability and controversiality of Covid-related Reddit posts and comparing their performance to decide on the final model to be used in our product.

Challenge and Target Audience

The extensive usage of freely available online social platforms such as Reddit has opened the door for many to amplify content freely. The attempts to reduce the spread of misinformation by fact-checking and flagging posts with inaccuracies may help reduce the influence of false information (Gaozhao, 2021). In light of the lack of content moderation compared with other platforms, we hope to provide a content analyzer tool, "Postly", for Reddit users who frequently post Covid-related information in Covid-related threads as a source of self-regulation to encourage them to post responsible content to prevent misinformation. In future, we hope to further enlarge the power of this tool by making it available on all Reddit threads to promote the

construction of a responsible and healthy online environment.

Product Overview

Features

Our application will analyze the post drafts and provide an acceptability and controversiality label with a focus on topics related to Covid-19.

• **Predict acceptability** We consider the acceptability of a post to be measured by the difference between the number of upvotes and downvotes, which corresponds to the definition of "score" on Reddit. Contents with a lower predicted score are more likely to attract dislikes when posted.

• **Predict controversiality** The controversiality label identifies the post as either controversial or not controversial. A post will only be checked against controversiality if the sum of upvotes and downvotes exceeds 50. Contents with a controversial label are more likely to incur disputes and disagreements among the audience of the posts. Words or phrases that contribute to a low score or a controversial label will be underlined for users' reference to improve the quality of the post.

User interface

When users are drafting posts, PostLy will analyze the content and underline words and phrases that are likely to contribute to low scores or controversiality (Figure 1a). The colour of the underline exhibits the severity of the expression, for example, a red underline means that the text is highly provocative. When users click on the alert button, a detailed analysis will be shown in a pop-out window (Figure 1b). The predicted score is one of the five ranges, "< -100", "-100 to -20", "-20 to 20", "20 to 100", "> 100", with an arrow pointing at the estimated position of the predicted score. The score estimates the difference in number of upvotes and downvotes, hence predicting the acceptability of the content by the public. The classification of controversiality suggests to posters whether they risk triggering disputes by posting the content. Suggestions on how to improve underlined texts will be provided in the detailed analysis as well.

Experiment Setup

We followed a standard flow to set up experiments, where we collected and sampled data, preprocessed data so that they fit our purpose and can be applied to various machine learning models, applied different models to the data and collected the results.

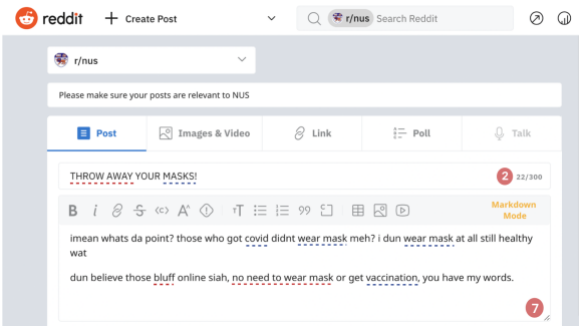

Figure 1a: Underlined texts by PostLy

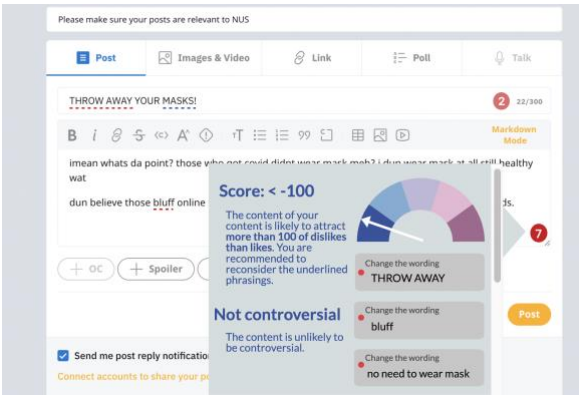

Figure 1b: Pop-out window by PostLy

Figure 1: User interface

Data Collection

In order to develop a tool to analyze Reddit's post quality, we collected a total of 204,573,086 raw Reddit posts through an online data source *Pushshift.io* which is publicly available. To ensure a fair distribution of Reddit posts, we adopted probability sampling to collect a sample set of posts from each month over a time span of 6 months from January to June in the year 2021. We noticed that some of the data lack key information that we were looking for. For example, some of the post body (i.e. the content) was deleted or removed by the administration, and hence we removed such data from our dataset.

As a post can belong to any multifarious topic group and its acceptability and controversiality might be affected by the topics themselves, it was essential to focus on a particular topic to carry out our studies. In this study, we selected Covid-19 as our focusing topics. We constructed a set of Covid-related keywords such as "covid", "coronavirus", "vaccination" and "quarantine", and filtered irrelevant content against this set.

Additionally, given that the target audience of our tool would be the frequent posters on Reddit, we limited the user ID in our dataset to those who have posted for more than 5 times over the span of 6 months with the help of Spark.

We further removed unnecessary columns of data objects such as “author flair”, “comment type” and kept 4 key attributes as shown in Figure 2.

Attribute Number	Attribute Name
1	Id
2	Body
3	Controversiality
4	Score

Figure 2: Four attributes kept during data preprocessing

In Figure 2, “Id” refers to the poster ID. “Body” contains the content of the post. “Controversiality” contains a binary value which represents whether the post is considered as controversial. Specifically, a post is labelled as controversial by Reddit if the ratio of number of upvotes and downvotes falls within a range from 0.4 to 0.6 and the sum of number of upvotes and downvotes is greater than 50. “Score” contains an integer which represents the differences between number of upvotes and downvotes.

Data Preprocessing

To prepare the data file for analysis, we performed cleaning and processing of the raw text data.

Data Cleaning

Data cleaning involves filtering of less important data and re-classification of the score attribute. While all posts in the dataset are Covid-related, not all of them have the same importance. For example, a post that receives 100 upvotes and 100 downvotes will have the same score as a post with no upvotes or downvotes at all. However, it is very likely that the former is a better and fair representation of public opinions on a specific post as the higher number of votes indicates a much higher number of views as well as a relatively unbiased pool of opinions. Hence, given the lack of an exact number of upvotes and downvotes, we further filtered our dataset by eliminating posts with both a controversiality of 0 and a score with an absolute value below the threshold of 50.

Meanwhile, the current representation of the score is an arbitrary number, where a small change in the number might not be significant to users. Hence, we decide to divide each score by 25 and use the rounded integer as our predicted value, so that a change in the value would mean a significant change in score. Overall, after the cleaning process, our original dataset is condensed into a new one containing 41865 posts.

Text Preprocessing

To further improve the quality of text data, we performed a series of text preprocessing steps:

- Removed all unwanted characters including emojis, user mentions, punctuations and hyperlinks.
- Removed all stopwords using NLTK library, which are the most frequent terms with the least significance for text processing.
- Tokenized text data and applied lemmatization to the tokens.

Feature Engineering

From the preprocessed text, we adopted a variant of the Bags-of-Words model, Term Frequency-Inverse Document Frequency (TF-IDF), where the value of a word increases proportionally to count, but it is inversely proportional to the frequency of the word in the corpus. Instead of a simple frequency count, the usage of IDF can help us remove the most frequently occurring words which do not contribute much to the analysis, such as “covid” in Figure 3. The use of TF-IDF vectorizer helped us vectorize the raw text data set for processing by the machine learning models.

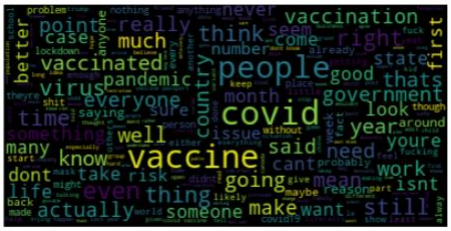

Figure 3: Word cloud generated from our dataset

Models

It is worthy to note that the score and controversiality of a Reddit post are variables of different nature. Score is an integer quantity that theoretically ranges from negative infinity to positive infinity, whereas controversiality is a binary value. Therefore, it is reasonable to apply different types of models on the two variables, namely regression model for score and classification model for controversiality. Here, we explored a total of 9 machine learning models and 3 deep learning models. We will choose the two with the best performance as the model to be used for our product. For each model, we choose 80% of the dataset as the train set and 20% of the dataset as the test set. We also select 20% from the train set as validation set when required.

Models Used to Predict Score

- **Linear Regression** Linear regression is a linear model which is used to determine if there is a linear correlation between parameters and the output score. We used LinearRegression from scikit-learn library in this project.
- **K-nearest Neighbor Regressor** K-nearest Neighbor (KNN) Regressor is used for regression where the function is approximated locally. KNN does not make assumptions on data distribution. It responds quickly to new training data

and is easy to use. We used KNeighborsRegressor from scikit-learn library in this project.

• **Recurrent Neural Network Regressor** Recurrent Neural Network (RNN) processes inputs with its internal state. In this project, we used Long Short-Term Memory (LSTM) since it can process entire sequences of data which corresponds to Reddit posts (Kowsari et. al. 2017). We also see a potential in RNN to give timely feedbacks while posters are drafting their posts. We built the RNN model by adding an LSTM layer within the model as well as other necessary layers such as Dense layers and Dropouts. Specially, since we are targeting at a regression problem, we added a regression layer as the output layer. We used LSTM from Keras library as the core of RNN model.

Models Used to Predict Controversiality

• **Logistic Regression** Though being a regression algorithm, logistic regression is suitable for modelling the classification of controversiality as a binary outcome and predict the probability for each outcome to occur. The default algorithm used is one-vs-rest. Among all solvers, *lbfgs* is chosen for its vigour in modelling unscaled datasets of smaller sizes. We used LogisticRegression from scikit-learn library in this project.

• **Decision Tree Classifier** Decision Tree (DT) undergoes supervised learning to provide a prediction for classification. DT is explored as it can handle categorical data and it uses a white box model to produce results that are easy to interpret. We prepared balanced datasets to avoid producing biased DT for this model. We used DecisionTreeClassifier from scikit-learn library in this project.

• **Random Forest** Random Forest leverages Bagging and Feature Randomness to generate a large number of relatively uncorrelated models. This classifier balances prediction error across individual decision trees, hence reducing variance and controlling over-fitting (Varoquaux et. al. 2015). We used RandomForestClassifier of scikit-learn library in this project.

• **Multilayer Perceptron** Multilayer Perceptron (MLP) conducts supervised learning using backpropagation and minimizes the cross-entropy loss function. It provides a non-linear function approximator for classification. L2 regularisation is applied to prevent overfitting. We used MLPClassifier from scikit-learn library in this project.

• **Light Gradient Boosting Machine** Light Gradient Boosting Machine (LGBM) is a Gradient Boosting framework faster than the conventional XGBoost. It produces a model typically based on an ensemble of decision trees. The quality of fit of each subsequent learner enhances by revising the classification errors of precious tree models. Logistic function is adopted as the loss function for this project to model the classification of controversiality. We used LGBMClassifier from lightgbm library in this project.

• **Complement Naïve Bayes** Complement Naïve Bayes (CNB) derives prediction by computing the complement of each class. It is selected for its robustness in modelling imbalanced datasets and suitability for text classification tasks (Frunza et. al. 2010). We used ComplementNB from scikit-learn library in this project.

• **K-nearest Neighbor Classifier** K-nearest Neighbor (KNN) Classifier is used for classification where the function is approximated locally. The algorithm is similar to the KNN Regressor which is mentioned earlier.

• **Support Vector Machines Classifier** Support Vector Machines (SVM) are supervised learning methods helpful for classification. We used LinearSVC from scikit-learn library in this project because it is efficient in supporting one-vs-rest multi-class classification for relatively small datasets. SVM is also useful in text categorization.

• **Recurrent Neural Network Classifier** Recurrent Neural Network (RNN) Classifier is similar to the RNN Regressor mentioned earlier except that the output layer is simply a classification layer.

• **Convolutional Neural Network Classifier** Convolutional Neural Network (CNN) Classifier is conventionally known as a useful way to do image identification because of its ability to assign different weights to different objects inside images. We can also apply this idea to text classification problems by identifying certain patterns in the text and assign weights to them so that an overall classification can be made. We used a Conv1D layer and several other Dense layers from Keras library to build the CNN model in this project.

• **Bidirectional Encoder Representations from Transformers** Bidirectional Encoder Representations from Transformers (BERT) and other Transformer Encoder architectures have been wildly successful on a variety of tasks in natural language processing (Jin, 2020). They compute vector-space representations of natural language that are suitable for use in deep learning models. The BERT family of models uses the Transformer Encoder architecture to process each token of input text in the full context of all tokens before and after. BERT models are usually pre-trained on a large corpus of text, then fine-tuned for specific tasks, which might be suitable for our purpose. We used the structure as in Figure 4 through the small BERT model from TensorFlow in our project.

Explainability of Prediction Result

In view that users might not be able to understand the numbers predicted by the models directly, we will convert the predicted score and controversiality into graphical representations (as shown in the previous section) and mark

the words from the Bags-of-Words that contribute to low score or controversiality, which is powered by the explainer built from the *lime* package. The use of explainer can enhance the explainability of results to suggest to users which word or phrase contributes to the prediction.

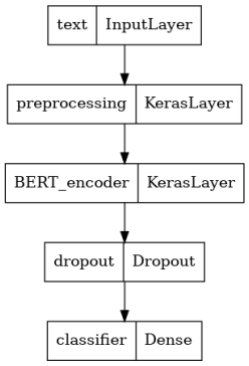

Figure 4: Structure of BERT used in this project

Results and Discussion

Score Prediction

Mean squared error (MSE) is a common quantity used to access the performance of a regression. A smaller MSE means a better performance. We compute the MSE for each of the three models and obtain the following table (Figure 5). From the table, we can observe that the MSE produced by RNN Regression is much smaller than that of Linear Regression and KNN Regressor. This could be reasonable because texts, when compared with normal data, contain more complex internal relations among the words and this may require more internal layers and complex algorithms (Kamath et. al. 2018).

	MSE
Linear Regression	670.377
KNN Regressor	390.442
RNN (Regression)	47.830

Figure 5: Mean squared error for each model

Furthermore, we compared the fitted value of score with the actual value of score by plotting a graph of fitted score against actual score (Figure 6). The performance of each model can thus be easily seen from the distribution of data points (red dots in the figure) with regards to the line $y = x$ (blue line in the figure). It is consistent from the MSE results that RNN Regression gives the most accurate prediction. Meanwhile, we also notice that texts with actual score closer to 0 are likely to have a worse prediction. This may be explained by the fact that a high score of a post implies a high post acceptability but the inverse might not be necessarily true.

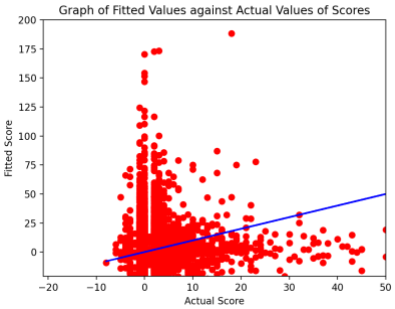

Figure 6a: Linear regression

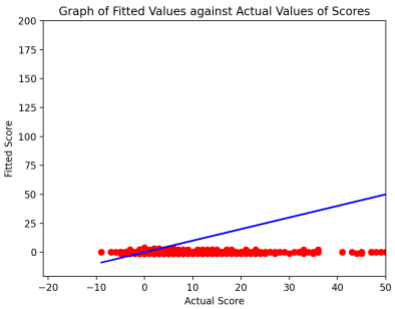

Figure 6b: KNN regressor

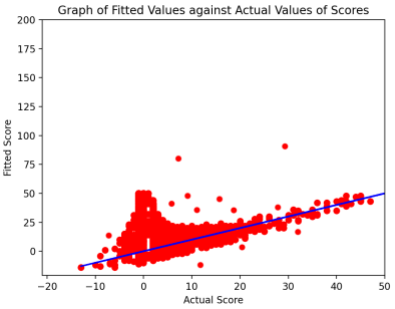

Figure 6c: RNN (regression)

Figure 6: Graphs of fitted score against actual score for each model

From the above results, we would choose RNN Regression as the driving core of our score prediction feature since it gives the best performance.

Controversiality Prediction

In classification problems, we use accuracy as a metrics of the performance of a classification algorithm. Accuracy refers to the ratio of correctly classified instances to the total number of instances. We compute the accuracy for each of the three models and obtain the following table (Figure 7).

From this table, it can be observed that most of the selected algorithms give an accuracy of around 65% to 70%,

hence we cannot make a decision purely from the experiment results. However, it has been proven that BERT, as a relatively new technique, outperforms the other techniques in the area of real-time text processing, for its ability to reuse pre-trained models and produce much better performances (Koroteev, 2021). Hence, we would choose BERT as the driving core of our controversiality prediction feature.

	Accuracy
Logistic Regression	0.680
Decision Tree	0.606
Random Forest	0.672
Multilayer Perceptron	0.662
LGBM	0.685
CNB	0.681
KNN Classifier	0.667
SVM Classifier	0.669
RNN (classification)	0.671
CNN (classification)	0.666
BERT	0.685

Figure 7: Accuracy for each non-regression model

Ethics

Posts used in train models are retrieved from public datasets with no personal information. Our project does not analyze users' content in any way except to provide predictions. We will not claim ownership of any content sent for prediction. Users' texts will not be used to train the model unless with users' consent. Texts selected for prediction will not be stored and will be deleted immediately when the prediction is completed. Users' texts will not be made available to the public or shared with other parties in any form.

Limitations and Future Work

As TF-IDF often ignores the word order, TH-IDF may not always be accurate in deriving the semantic meaning in the given context. Hence, the usage of rhetorical devices such as double negative may affect the output. The use of BERT will likely produce better output as it does consider the context of the words. However, the use of BERT would be more computationally expensive (Liu et. al. 2020). We may want to utilise the more efficient variants of BERT in future. Moreover, BERT did not outperform the other techniques by much in our preliminary experiments, hence we would need to conduct more experiments with different and larger datasets in future to prove its practicability.

Meanwhile, at the current stage, our analyzer focuses on Covid-19 contents as it has been one of the most popular and influential topics over the past two years. However, we may see a decrease in Covid-19 related discussions overtime. We hope to further increase the utility of our analyzer by expanding it to different trending topics, which will evolve with time. We would like to construct a streaming pipeline to collect new post data from Reddit streaming APIs to make our classification more robust.

Conclusion

This report has shown the practicability of using machine learning and deep learning tools to predict the scores and controversiality of a post simply from its content. We compared different models by their performances on score prediction and concluded that RNN Regression is the most suitable model for score prediction. On the other hand, our experiments on controversiality prediction did not give a clear trend and we decided to use BERT for its various proven advantages. We would be applying these two models on our product, PostLy, which will translate the predicted data into symbols and human languages so that users can have a better understanding of the feedbacks and suggestions made by PostLy. Further studies can be conducted on whether the chosen models have the potential to performed well on real-time texts as users type their drafts.

References

Frunza, O.; Inkpen, D.; and Matwin, S. 2010. Building systematic reviews using automatic text classification techniques. In Coling 2010: Posters (pp. 303-311).

Gaozhao, D. Flagging fake news on social media: An experimental study of media consumers' identification of fake news. *Gov. Inf. Q.* **2021**, *38*, 101591.

Jin, D.; Jin, Z.; Zhou, J.T.; and Szolovits, P. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 8018-8025).

Kamath, C.N.; Bukhari, S.S.; and Dengel, A. 2018. Comparative study between traditional machine learning and deep learning approaches for text classification. In Proceedings of the ACM Symposium on Document Engineering 2018 (pp. 1-11).

Koroteev, M.V. 2021. BERT: a review of applications in natural language processing and understanding. arXiv preprint arXiv:2103.11943.

Kowsari, K.; Brown, D.; E., Heidarysafa, M.; Jafari Meimandi, K.; Gerber M. S; and Barnes L. E. 2017. HDLTex: Hierarchical Deep Learning for Text Classification. 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 364-371, doi: 10.1109/ICMLA.2017.0-134.

Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Deng, H; and Ju, Q. 2020. Fastbert: a self-distilling bert with adaptive inference time. arXiv preprint arXiv:2004.02178.

Mohan, S.; Guha, A.; Harris, M.; Popowich, F.; Schuster, A.; and Priebe, C. 2017. The Impact of Toxic Language on the Health of Reddit Communities. In: Mouhoub, M., Langlais, P. (eds) Advances in Artificial Intelligence. Canadian AI 2017. Lecture Notes in Computer Science(), vol 10233. Springer, Cham. https://doi.org/10.1007/978-3-319-57351-9_6.

Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; and Mueller, A. 2015. Scikit-learn: Machine learning without learning the machinery. GetMobile: Mobile Computing and Communications 19(1): 29–33.