Tweets Analysis on Singapore General Election 2020

Thu Ya Zaw Naing [1 2] **Li Xiaoyin** [1 3] **Mengxue Ou** [1 4]

Abstract

On July 10, 2020, the results of the 13th General election of Singapore were announced. The People's Action Party (PAP), which has been in power for 55 consecutive years, won the election again with a lower vote percentage than the last election. As one of Singapore's most significant events, a heated debate has arisen on social media like Twitter. In this project, tweets extraction is conducted to collect tweets concerning Singapore General Election. Different text mining techniques, including sentiment analysis and topic modeling, are utilized to acquire insights from extracted tweets. Code and dataset are available at https://github.com/thuyazawnaing/K6312-GE2020SG

1. Introduction

Nowadays, with the advent of the Internet, people can express their opinions on social issues on different social media platforms. In other words, social media is closely connected with Internet users. This situation provides a great chance to explore insights from social media texts using up-to-date machine learning techniques.

1.1. Singapore General Election 2020

The SG2020 General Election was held under the Covid-19 pandemic. On July 10, 2020, the results of the 13th General election of Singapore were announced. The People's Action Party (PAP) has been in power for 55 consecutive years, won 83 seats out of 93 seats, with an overall turnout rate of 61.24%. In 2020 PAP was re-elected successfully, but the voter turnout dropped by 8.66% and won 28 out of Singapore's 31 constituencies, which is one less than the previous general election.

A total number of 192 candidates from 11 political parties was vying for 93. Although the PAP can continue to govern, the opposition Workers' Party (WP) has achieved its best performance ever. WP not only held on to Hougang Single constituency and Aljunied Group Representation Constituency (GRC) but also captured the Shing Kong GRC with its seats in Parliament increasing from 6 to 10; in addition to. All the other nine opposition candidates lost in the selection.

GE2020 was a very intense campaign; the result was somewhat surprising. This was the first time that the ruling party, PAP, had its lowest votes since 2011. It's also the first time that the opposition party has made the largest representation in Parliament since 1966.

1.2. Social media's influence on the Singapore political landscape

The general election system rules remained the same as before, but due to the Covid-19, this year's general election promotion in Singapore had undergone unprecedented changes. The government provided more subsidies as well as Internet-connected venues for candidates to record live webcasts with unlimited airtime and duration. Under the Covid-19 safety measures, more than five gatherings are prohibited; most of the political campaign held online and reached out to voters via social media. PAP would not win the election without the support of young people. All the parties aware of it's very important to get young voters' support because they are Singapore's future. Due to the instantaneous popularization of social media, young-adult Singaporeans who were previously apathetic to politics are more engaged in politics than ever. Social media is transforming Singapore's political landscape.

In this general election, there are about 2.65 million voters in Singapore. According to statistics, 9 out of 10 Singaporeans have smartphones, and about 10% are 21 to 25 years old. The group of young-adult people who grew up in globalization is better at using social media to acquire pieces of information and voice opinions. During the GE2020, the fierce competition between the PAP and the opposition Parties also appeared on social media. As a result, PM Lee Hsien Loong posted his updates almost every day on his Facebook, Twitter. Supporters and opponents of the opposition party and the ruling party were reluctant to lose the prospect to voice their opinions on social media.

[1]Wee Kim Wee School of Communication and Information, Nanyang Technological University, Singapore [2]G1901249K [3]G2003071F [4]G1901048B.

In a nutshell, social media can motivate voters. Political competition in social media may also promote the enthusiasm of cybernauts to participate in political activities. In addition to commenting on the election on social media and debating with opponents, this generation of young voters also shares their opinions with their family, relatives, friends. To some extent, it may also impact the older family members and even drive the new political climate for the next generation of Singaporean voters, who often use social media. Thus, social media can extend its influence on reality through interpersonal communication. Besides, young people are more familiar with social media. It is beneficial for opposition parties because they can promote their platforms at low costs, bring their party's concerns to the forefront through social media, and influence the election in a way that makes their voices heard. This made social media a rich source of information for academic researchers to perform information mining and sentiment analysis.

2. Related Work

Information mining on social media has been long a popular topic. The research on this topic has never stopped. Over the decades, many advancements have been made on similar projects. Early in 2010, Twitter had been used to conduct political, sentimental analysis. The research conducted a content analysis of over 100,000 messages containing a reference to either a political party or a politician, and the results showed that Twitter is indeed used extensively for political deliberation(Tumasjan et al., 2010). Later, the tweets were even used to predict the election. One research conduct a prediction on the US Massachusetts election. By annotating tweets annually, the work use sentiment lexicons to classify texts into positive, negative, and neutral(Chung & Mustafaraj, 2011). Another work used a model combining Tweets popularity and sentiment analysis. Considering the influence of the party, classifier SVM was used to build a sentimental model(Bermingham & Smeaton, 2011). Moreover, some researchers automatically compiled a 2012 US presidential election tweet dataset and described two automatic systems that predict emotion and purpose in tweets(Mohammad et al., 2015). In 2018, one paper studied information diffusion in social media and the bots' role in shaping public opinions. The results suggested that aggressive use of Twitter bots, coupled with social media fragmentation, could contribute to the vote outcomes(Gorodnichenko et al., 2018).

With the high veracity of data, to uncover the hidden insights, various data mining techniques are also paramount. Use of Latent Dirichlet Allocation (Mehrotra et al., 2013) is very popular approach to uncover the hidden topic but (Nur'aini et al., 2015) introduces the use of singular value decomposition with the combination of Kmeans clustering produces sensible results which is an interesting area of exploring topic modelling. Techniques such as word embeddings and convolutional neural network(CNN), Long Short Term Memory networks (LSTM) (Hochreiter & Schmidhuber, 1997) have attracted extensive attention in sentiment classification tasks. Experimental results showed that using word embeddings, CNN, and LSTM leads to statistically significant improvements over various baselines(Yang et al., 2018). The same methods are also used in other research work(Dorle & Pise, 2018). In recent years, social media has hailed as a vehicle and tool of political manipulation(Badawy et al., 2019).

3. Data Preprocessing

3.1. Data Extraction

We focus on the available tweets related to Singapore General Election 2020 on Twitter as the source of data. Different from the current prevalent method to use the Twitter API and Python Tweepy package to scrape tweets from Twitter, we used the Python Selenium package to extract information from the Twitter webpage automatically. As the standard Twitter API that we have applied from Twitter only allows us to retrieve tweets up to 7 days ago and has limitations on the number of tweets scrapped (i.e., no more than 18,000 tweets every 15 mins). We ultimately decided to use the Selenium package on Python Version 3.0 to extract the data, permitting us to scrape data regardless of the time and number limitation. Considered as a trendy open-source and web-based automation tool, Selenium can scrape a large volume of data such as texts and images from the website within a relatively short time. It can also automatically simulate human web browsing behaviors and patterns (e.g., scrolling). To be specific to our data scraping process, firstly, the Selenium package and Chrome driver were installed and imported on the Notebook. A google chrome instance was then created, which allowed the program to open an URL in google chrome and open the website that we scrapped. Third, the XML code of the information we are interested in was identified by looking for the XPath of the website's information. The functions of "find_elements_by_xpath" and "get_attribute" were used to get the elements and attributes of interest. A series of keywords were used to search for the tweets related to Singapore General Election 2020 on Twitter, which generated different URLs for us to scrape the data. These keywords include "GE2020SG", "GESG2020", "PAPSingapore," "SGElection," "SGGE," "Singaporevotes," "SingaporeVotes," "general_election_sg2020," "sgelections," "wpsg," "Ge2020," "GE2020," "Singapore GE 2020," "Singapore general election." The scraping process finally yielded a total of 9,068 tweets relevant to the issue of the Singapore General Election 2020. Though we extracted the tweets' information, including the Users' ID, time of tweets, the content of tweets, the number of comments, likes, and retweets, we only focus on the content of tweets in our data analysis.

3.2. Text Preprocessing

After collecting tweets' text, data cleaning is performed to remove corrupt and futile data. Generic cleaning of data is conducted by removing the tweets in which text contents are less than three words long. Some specific NLP tasks are performed as part of the preprocessing activity to derive meaningful insights from our data, including

- **Remove Punctuation**: Removing punctuation from the textual data because they do not contain any useful information.
- **Remove hashtags and user names**: Hashtags and usernames are typical formats of tweets. Hashtags, which are usually seen as '#,' indicate the tweets' topic, while usernames, which are generally seen as '@,' show the user this tweet wants to mention or refer to. These hashtags and usernames influence the modeling building accuracy so that we remove them.
- **Stopwords Removal**: Data is extracted from irrelevant characters like the commas, semicolon, full stop, double quotes, brackets, special characters, etc. Data is assembled free from those so-called stop words which appear customarily in the textual data. This removes terms that provide little information about the text content. To make the tweets analysis more reliable, we come out with a list of extended stop words after investigation. We then created a custom_stopwrods.txt file, which consolidates all the stopwords declaration.
- **Word Stemming**: After removing stop words, the next step is stemming. This step decreases a word to its root. The rationale behind stemming is to clear the suffixes and bring down the number of words. For instance, words like "collaboration," "collaborated," "collaborating" can all be transformed into the single term "collaborate." This can save time and reduce the computational complexity for subsequent analysis. Among many popular stemmers like Lovins Stemmer, S-Stemmers, Porter Stemmer, Paice/Husk Stemmer, we chose SnowballStemmer, the most commonly used stemmer.
- **Tokenization**: Chopping up sentences into individual terms known as tokens.

3.3. Data Annotation

We need label data for sentiment analysis, but conducting manual annotation work for the dataset is labor-intensive and time-consuming. We use existing sentiment analysis tools to get the label for our data to hedge this challenge. From there on, we can build our model to conduct sentimental analysis and text classification to derive the insights. To annotate the dataset, we employed the following techniques to generate the label:

- **TextBlob**: A library is used to generate the sentiment score from the textual data. And it gives the single sentiment polarity scores range from -1 to 1.
- **VADER Sentiment Analyzer**: A pre-trained model in the NLTK python package which uses rule-based values tuned to sentiments. The analyzer provided the sentiment polarity scores range from -1 to 1. The polarity scores are set as 0 for neutral, 1 for positive, and 1 for negative.
- **Ground Truth**: We then take the mean of the two analyzers' outputs, derive the overall sentiment scores, and defined the label.

Figure 1 shows the final output of text annotation, which is used as our Ground Truth.

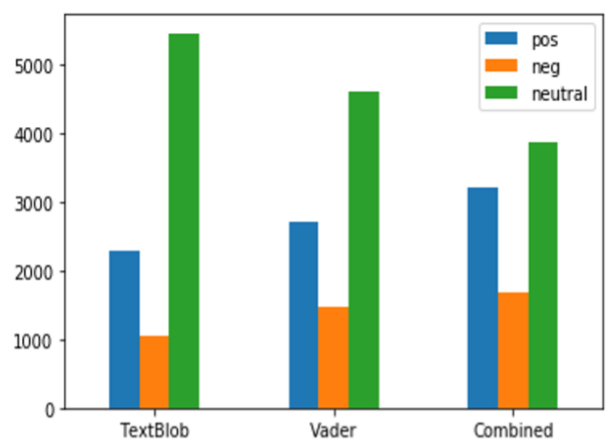

Figure 1. Ground Truth Distribution

4. Exploratory Data Analysis

After we extracted the tweet's data, the next step is to conduct exploratory data analysis to see what insights can be acquired from the extracted data and how we can perform research. In this part, some text analysis techniques like text preprocessing, frequency count, and visualization techniques such as word-cloud support the analysis process.

4.1. Overview of All Tweets

In the Exploration stage, once we have the unstructured text preprocessed, we looked at word occurrences as well as their frequency distribution to discover any underlying information or concepts. In this part, a brief overview of tweets is conducted to show the distribution of tweets. As shown in Figure 2, tweets posted in July constitute almost 70% of all the collected tweets, approximately six times more than those in May and June.

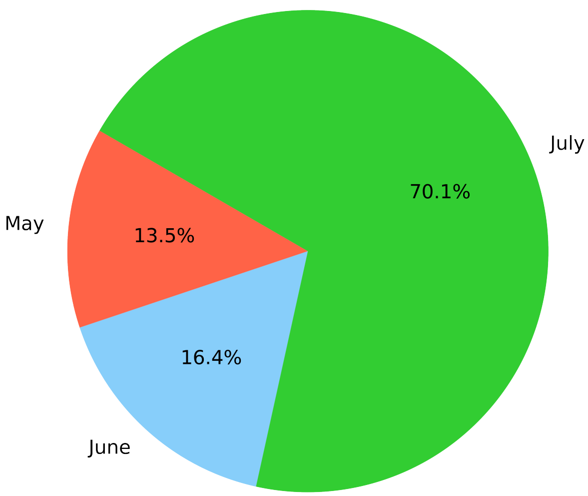

Figure 2. Tweets number by months

Figure 3 shows the number of tweets posted on different dates in July. The tweets are concentrated in July. It can be found that the nearer to the election date, the hotter the topic and the more tweets posted. However, the number of tweets declined sharply as the General Election ended.

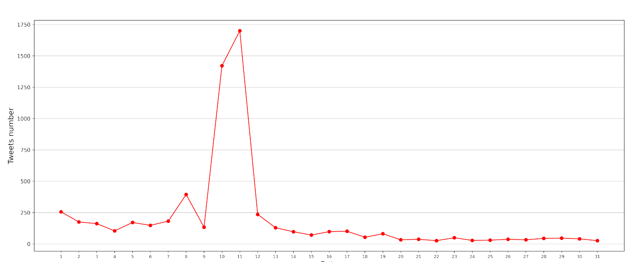

Figure 3. Tweets number by date

4.2. Tweets Visualization

It's hard to uncover hidden insights from a large number of texts. However, visualization can make this work easier. In this part, text mining techniques like frequency counts and wordcloud are used to acquire tweet texts' insights.

Figure 4 is a dispersion plot of words "singapore," "election," "pap," "wp," which shows the distribution of words in the texts. From this figure, it can be easily seen that "Singapore" and "election" frequently appears from beginning to the end, which shows the topic of tweets. The word "pap" is also distributed densely, followed by "wp," "psp" and other parties. Overall, PAP is the party with the most concern.

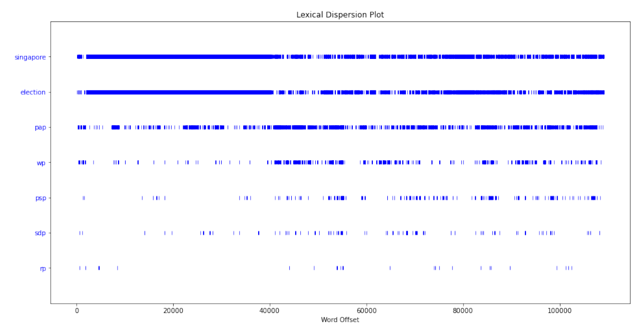

Figure 4. Lexical dispersion plot

Bigram is also one of the most important text mining methods, which focus on the allocations of certain words. Figure 5 states the bigrams of the word "pap." Among them, "win" is the most frequent word. In reality, PAP also won the election; it is fascinating to see our findings in line with it.

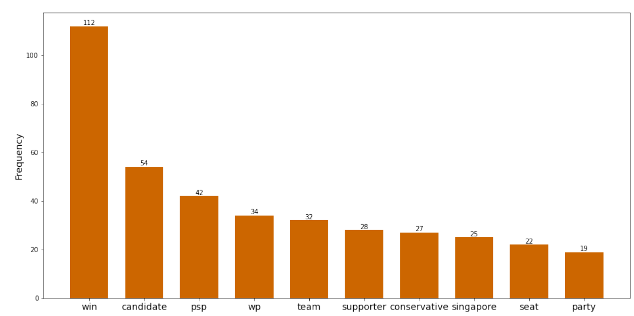

Figure 5. pap-bigram

Figure 6 states the bigrams of the word "wp." Among them, "pap" is the most frequent word. It seems that people on Twitter like to compare WP with PAP. Besides, words like "win," "Sengkang," "east" also appear, reflecting that WP won the vote in the Sengkang area with its location in the northeast of Singapore. In general, the real election situation and result is consistent with the visualized pictures and tweets.

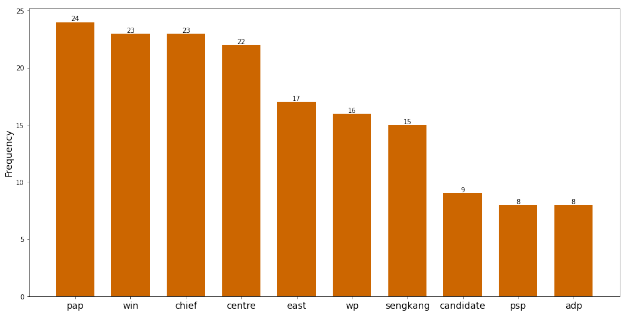

Figure 6. wp-bigram

The following figures(Figure 7, Figure 8, Figure 9) present the word clouds of tweets with negative, positive, and neutral sentiments. All the word clouds contain key words "Singapore and election," indicating these tweets' topic. Obviously, PAP is the most popular party with the most concern, while Twitter users conclude other parties as "opposition". In positive tweets, PAP frequently appears, which explains the final winning of the election of PAP. In negative tweets, PAP also appears often, which can explain the drop in voter turnout of PAP. In neutral tweets, people seem to be more interested in the election itself and the final result, as words "vote", "count", "result" are frequently used. Unlike positive and negative tweets, words in neutral also appear to be impartial. In general, the word clouds reflect the real situation in the General Election.

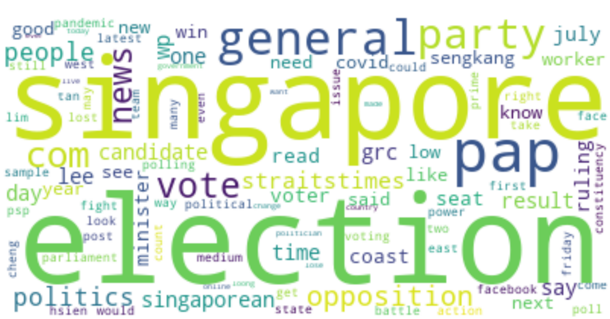

Figure 7. wordclound of negative tweets

5. Model Building

5.1. Topic Modeling

Topic Detection is the method of detecting topics the evolution of those topics in a continually updated pool, there-

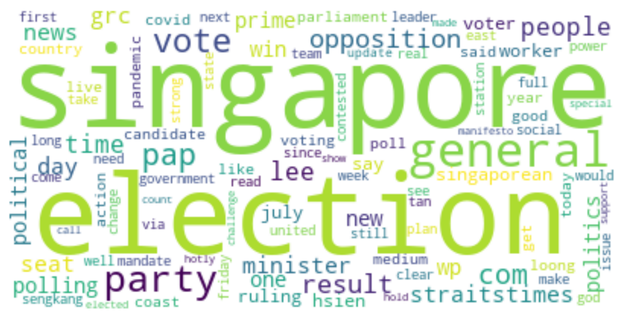

Figure 8. wordclound of positive tweets

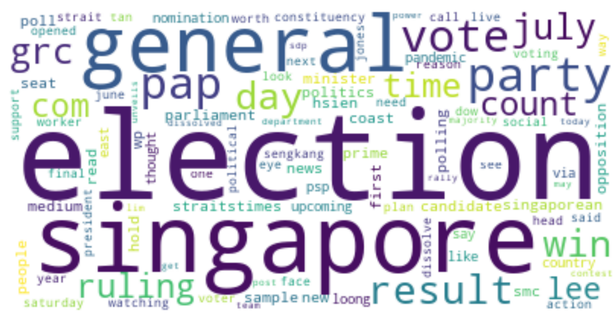

Figure 9. wordclound of neutral tweets

fore in our Modeling Stage, we used the preprocessed text and created a Term-Document Matrix based on the Bag-of-Words model and utilized unsupervised methods like Clustering and Topic Modelling using LDA to discover hidden themes amongst the dataset. From the resulting Clusters and Topics discovered by the LDA algorithm, we then look at the documents associated with each topic to identify common contents and derive possible labels for these documents. We then use clustering techniques to address the following questions.

- Q1: What topics do people's tweets cover? Can we separate them into a few categories?
- Q2: Within each topic, what events or issues are discussed?
- Q3: How can we identify the category of tweets automatically when it is submitted?

In our scenario, clustering is performed on the tweet's content using the K-Means clustering technique. With clustering, contents are group accordingly so that the same group of contents is more homogeneous than others. Ideally, this approach divides the contents into clusters k; each content belongs to the center of the nearest cluster, i.e., the mean of their content. Thus, we interpret the cluster centers as topics of the content. Clustering high-dimensional data like tweet's content is a computationally intensive task. Singular Value Decomposition (SVD) is a famous method for dimensionality reduction for textual data, also called latent semantic analysis (LSA). With dimensionality reduction, we can reduce the dataset's dimension to a smaller size while enhancing time complexity. We combine K-means clustering and Singular Value Decomposition as a topic detection technique in tweets' content. Firstly, the tweet content's dimension is reduced using SVD. And apply K-mean clustering on these reduced forms. The topic of the cluster is formed by the centroids of those clusters. Our investigations show that SVD and K-mean clustering's hybrid approach gives promising comparative accuracy. And it is also on par with the technique not using dimensionality reduction with faster computation time. "Truncated SVD" algorithm is applied to our dataset with a total of 3 components. Starting from eliminating both standard and custom stop words and following by removing the non-alphabet and special characters. Next, term frequency-inverse document frequency weighting scheme is used to weight the matrices, and we perform the following in a sequential manner.

- Vectorize the words into token using TF-IDF Vectorizer
- "Tweet" matrix's dimension is reduced with the use of Singular Value Decomposition
- Using K-means clustering, group "Tweet's content" in the reduced form
- Transform back the uncovered centroid into the original dimension, i.e., words
- A specific topic is formed by the top-weighted words in each centroid

The optimal number of clusters (k) is evaluated using the metric called "silhouette coefficient," and we choose k clusters in our case with 10% variance of SVD step and "silhouette coefficient" is used as metric for cluster's metric. Detailed analysis will be discussed in Section 6.1.

5.2. Sentiment Analysis using Text classification

Another main area of this project is sentiment analysis. For sentiment analysis, we partition the records into two parts: the training set and testing set, which are used correspondingly to train the classification models and validate the models. We used both machine learning and deep learning models to build our text classification model. Our overall sentiment analysis framework is shown in Figure 10.

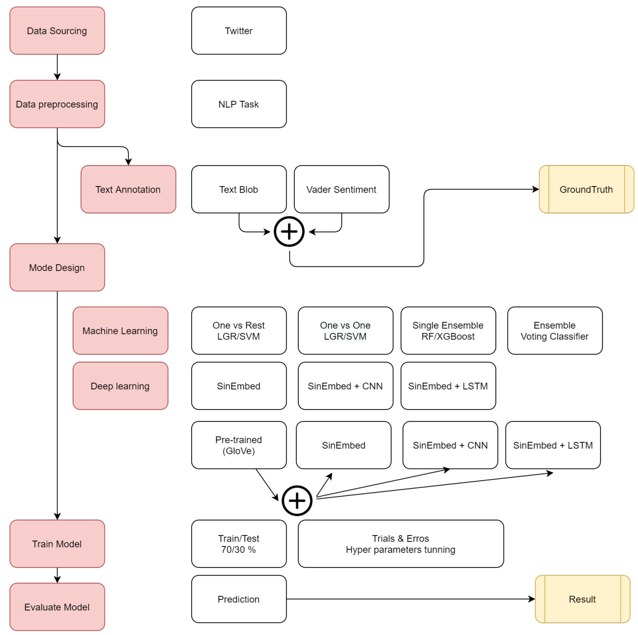

Figure 10. Sentiment Framework

5.2.1. Machine Learning

Before we build the classification model for text classification, the preprocessing for text data must be done, which removed some useless or unnecessary features (words). In Machine learning, we conduct a two-level approach of ensemble learning. Firstly we build the five machine learning models; Logistic Regression, Support Vector Machine, Naive Bayes, Random Forest, and XGboost. Since our problem is multi-classification, we also incorporate meta-strategies for logistic regression and support vector machine, i.e., "one vs. one" and "one vs. rest/all." All the base classifier is supervised machine learning which output is based on probability assumptions in which all the features (in case of text classification, the features are words) are assumed to be independent of each other, and each feature has no relation with others in the same sentence. Preprocessed data are fitted into the model for text classification to count each feature's frequencies and calculate each feature's contributions to the model's different classes. We then perform ensemble stacking, which uses base classifiers' predictions as input for training to a second-level ensemble model using an ensemble voting classifier. Detailed evaluation will be discussed in Section 6.2.

5.2.2. Deep Learning

For our deep learning model, we conduct two experiments. The first step is we build the model from scratch. The second experiment uses the pre-trained model to initialize the weight of our deep learning model. For experiment one, we construct three different deep-learning models, as listed below.

1. Single Embedding Layer: We make use of word embeddings based on the word-to-vector concept by Google to represent our tweet's content. These embeddings represent each of the words as vectors, which can then be used to train a neural network. Our multi-layer neural network comprised 40 nodes as input layer and three nodes representing three classes as the output layer.

2. Single Embedding Layer + convolutional neural network: We also tried to stack a convolutional neural network on the single embedding layer. Our CNN network comprised 128 nodes with the kernel stride of five, and a fully connected layer is mapped to the output of the CNN layer and followed by the output layer.

3. Single Embedding Layer + LSTM: As another approach, 128 memory units of the Long Short Term Memory networks layer are stack on top of the embedding layer. Then fully connected layer is mapped to the output of the LSTM layer. Lastly, a fully connected layer maps to three dense nodes representing three classes of the output layer.

The activation functions use in the input and output layers were relu and softmax, respectively. Categorical cross-entropy function was used to calculate the loss at each training epoch.

Transfer Learning: Pretrained NLP-Model were obtained from Stanford NLP, namely "glove.twitter.27B.100d.txt (http://nlp.stanford.edu/data/glove.twitter.27B.zip)" (Pennington et al., 2014), built on Twitter data. We employ Word2vec represents each word in a corpus as a vector in a vector space, where words with similar meaning would be closer to each other in the vector space. For experiment two, we use this pre-trained model to initialize the weight of the aforementioned three models.

Hyperparameter Tuning: As part of the attempts, we also tried the hyperparameter-tuning process of searching the ideal model architecture. Optimizers such as "Adam," "rmsprop," and "SGD" have been explored; however, in our case, "rmsprop" gave better accuracy.

Prevent Overfitting: We tried the epoch range from 20 to 200; however, the Early Stopping criteria is declared in the callback of the training, which can help to prevent the model from overfitting. Another approach to preventing the overfitting is that we also set the dropout to a 0.1 and spatial drop out as 0.1 for the LSTM model. Detailed evaluation will be discussed in Section 6.3.

6. Evaluation

6.1. Result Interpretation of Topic Modeling

The topic names and the words which feature prominently in each topic are shown below: From Figure 11, the topic is labelled as"PAP manifesto" and covid related issues for the People's Action Party due to the prominent of the words like "PAP" + "manifesto"+"covid"+"hygiene"+"safety" in the LDA output. The visualization of the wordcloud is generated from the frequency distribution of the words after data cleaning.

Figure 11. People's Action Party

We also try to group the content related to "Worker's party" into two clusters. Figure 12's legend distribution shows the most prominent topics are "I Stand With Raeesah Khan" and the second topic is about the opposition leader, "Pritam Singh" given by the LDA model. It's also fascinating to see the "Aljunied and Hougang" is very significant in the wordcloud, which truthfully reflects the real election results. Work's party won the seat in both "Aljunied and Hougang GRC."

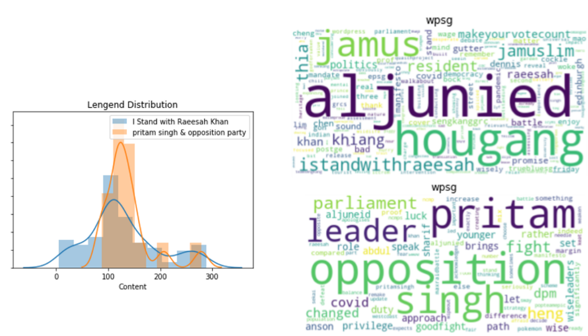

Figure 12. The Worker's Party

We further explore the whole Singapore General Election 2020; topics identified pertain to three diverse groups of

"No blank cheque - vote PAP out," "Politics and Covid pandemic," and the "Singapore Ruling party and PM Lee" given by the LDA model shown in Figure 13.

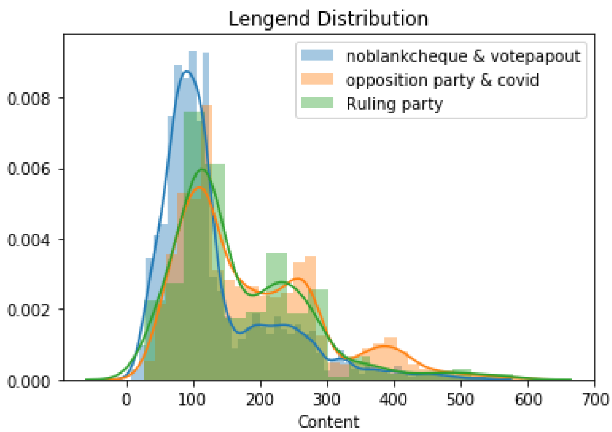

Figure 13. Singapore General Election 2020 : Topic Modeling

We can also visualize the most important factors in each topic by plotting most frequencies words using the word cloud shown in Figure 14

Figure 14. Singapore General Election 2020

6.2. Comparison of Machine Learning Models

As we set our baseline to achieve 88% accuracy, we first build our model using the feature size of 2000 and can't beat the baseline even with the use of ensemble learning. We then further increased the feature vector size to 5000. Figure 15 compares machine learning models with feature vectors of 2000 and 5000, respectively. We can conclude the Ensemble voting classifier with 5000 feature vectors outperform all the models with 89.13% accuracy.

6.3. Comparison of Deep Learning Models

Figure 16 shows the Embedding layer + CNN model without initializing the pre-trained model's weight outperforms all

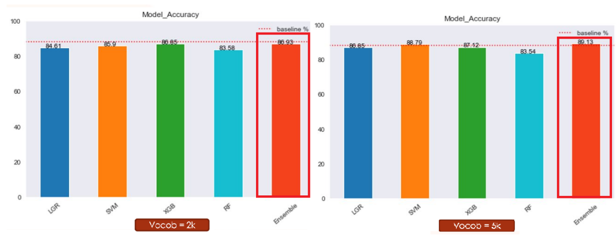

Figure 15. Machine Learning Accuracy

the models.

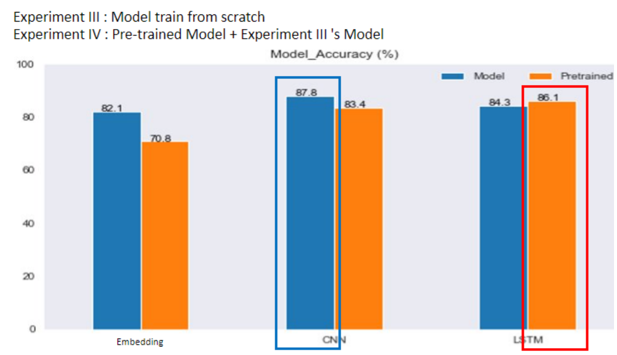

Figure 16. Deep Learning Accuracy

We also conducted the model evaluation by predicting the sentence, which is randomly chosen from the test set. Figure 17 shows the three scenarios of text predicted by the Embedding layer + CNN model without initializing the pre-trained model's weight and Embedding layer + CNN model with the initializing the pre-trained model's weight.

For case three, given ground-truth from the combination of "Vader" and "Textblob" is "Neutral," but both models predicted as "Negative." It's quite subjective to judge the label in this case. In my opinion, I probably will label it as "Negative." Overall we found these two models gave the same predictions in all three cases and shows its consistency in prediction.

7. Conclusion

In a nutshell, our project can be further improved in three major areas as listed below.

1. **Fine-tune every part of the system:** The machine learning model can be further improved by finetuning. For instance, XGBoost and Random Forest parameters are chosen using the Randomized-search due to the Gird-search being too time-consuming. Perhaps we

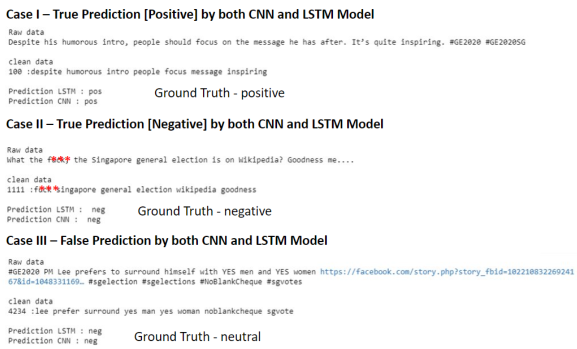

Figure 17. Result Interpretation of Deep Learning Model

can spend more time finding the best hyperparameters.

2. **Annotation quality:** The quality of the ground truth will limit any machine learning model. Current ground truth is derived from VaderSentiment and Textblob. To achieve better accuracy with sensible results, qualitative ground truth is required. It can be annotated by the human to increase accuracy while the workload will increase, or probably semi-supervised learning could be an alternative way to hedge this problem.

3. **Imbalanced dataset issue:** Oversampling vs. Undersampling methods can be explored, or we can extract more tweets related to GeneralElection and try to balance the dataset.

All in all, sentiment analysis and election prediction have always been a hot topic. Our project can also apply to other countries, for example, US election prediction and analysis. With machine learning and deep learning methods, the prediction will be more accurate, and sentiment analysis will also be easier to acquire.

Ethnical crawler: As mentioned in our data extraction section, Web crawling is the first step in our data mining process; deciding what is ethical in data crawling is entirely subjective and controvertible; however, our crawler serves as an "Academic Researchers' Crawler" and is designed to behave ethically in a pragmatic manner. We also would like to claim that both the data and the trained model will only use for academic research purposes

References

Badawy, A., Lerman, K., and Ferrara, E. Who falls for online political manipulation? In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, pp. 162–168, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366755. doi: 10.1145/3308560.3316494. URL https://doi.org/10.1145/3308560.3316494.

Bermingham, A. and Smeaton, A. On using Twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pp. 2–10, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/W11-3702.

Chung, J. E. and Mustafaraj, E. Can collective sentiment expressed on twitter predict political elections? 2011.

Dorle, S. and Pise, N. Political sentiment analysis through social media. In *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 869–873, 2018. doi: 10.1109/ICCMC.2018.8487879.

Gorodnichenko, Y., Pham, T., and Talavera, O. Social media, sentiment and public opinions: Evidence from# brexit and# uselection. Technical report, National Bureau of Economic Research, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Mehrotra, R., Sanner, S., Buntine, W., and Xie, L. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR*, 2013.

Mohammad, S. M., Zhu, X.-D., Kiritchenko, S., and Martin, J. D. Sentiment, emotion, purpose, and style in electoral tweets. *Inf. Process. Manag.*, 51:480–499, 2015.

Nur'aini, K., Najahaty, I., Hidayati, L., Murfi, H., and Nurrohmah, S. Combination of singular value decomposition and k-means clustering methods for topic detection on twitter. In *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 123–128, 2015. doi: 10.1109/ICACSIS.2015.7415168.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. volume 14, pp. 1532–1543, 01 2014. doi: 10.3115/v1/D14-1162.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Fourth international AAAI conference on weblogs and social media*. Citeseer, 2010.

Yang, X., Macdonald, C., and Ounis, I. Using word embeddings in twitter election classification. *Inf. Retr.*, 21(2–3):183–207, June 2018. ISSN 1386-4564. doi: 10.1007/s10791-017-9319-5. URL https://doi.org/10.1007/s10791-017-9319-5.