Automated Pipeline for Sentiment Analysis of Political Tweets

* *

Atrik Das
School of Comp Sci. and Eng.
Nanyang Technological University
Singapore
atrik001@e.ntu.edu.sg

Kushal Sai Gunturi
School of Comp Sci. and Eng.
Nanyang Technological University
Singapore
gunt0004@e.ntu.edu.sg

* *

Aditya Chandrasekhar
School of Comp Sci. and Eng.
Nanyang Technological University
Singapore
aditya021@e.ntu.edu.sg

Abhinandan Padhi
School of Comp Sci. and Eng.
Nanyang Technological University
Singapore
abhinand001@e.ntu.edu.sg

Qian Liu
School of Comp Sci. and Eng.
Nanyang Technological University
Singapore
liu.qian@ntu.edu.sg

* Indicates Equal Contribution

***Abstract*—Social media such as Twitter have proven to be great resources of information regarding many events that proliferate the entire world. It also has the power to change the opinions of millions which is especially useful to sway the masses during political campaigns like presidential elections. Sentiment analysis can be performed on tweets to determine how people feel about certain political events which can be used to predict the behavior of people and propagate real-time change as the events play out. Therefore we decided to create a sentiment analysis and polarity detection pipeline generalized for all political data. The pipeline attempts to automate all the NLP tasks from data scraping to cleaning and pre-processing the dataset to make it ready for the classification tasks. The predictions are visualized via word clouds and a map color coded to reveal the sentiments of key nations around the world regarding the political event. This pipeline is tested from end-to-end with our own personal use case being to determine which candidate do countries around the world preferred during the 2020 US Presidential Elections: Trump or Biden. The pipeline provides fruitful results with a test accuracy of 73.73 percent.**

***Index Terms*—Political pipeline, Twitter sentiment analysis, Deep learning**

I. Introduction

We live in a world where data is pervasive to the majority of the decisions we make. Most data around us is usually unstructured and textual in format. This led to the rise of Natural Language Processing (NLP) which deals with finding meaning and patterns from seemingly unrelated text corpora [1]. One heavily researched area is sentiment analysis which refers to the classification of subjective opinions based on their attitude to the subject at hand [2]. This analysis is useful as it has proved to be beneficial in the realms of social media marketing and political campaigning. The former uses customer sentiments to determine how relevant their products are in their market demographic [3]. The latter uses big data analysis to mobilize voters during elections and influence public propaganda to maximize the political candidate's voting turnout [4].

Thus it is evident that knowing the sentiments of key nations around the world can be very beneficial while a political event with a far-reaching scale is playing out. In the highly connected world we live in today, Twitter has risen up to be the popular choice of social media [5] for the general populace to have healthy debate about political matters. Hence, researchers have often scraped tweets from Twitter to determine a country's sentiments about a certain topic [6]. However, after a comprehensive session of literature review [7] we found out that most of these political sentiment analysis publications are fairly similar in terms of methodology. For example, subjectivity detection and topic detection has been done to analyze people's sentiments about Brexit [8] and political leaning of Twitter users has been predicted by analyzing multi-party contexts [9]. This got us thinking that perhaps a generalized pipeline can be made to account for any sort of far-reaching political event that affects the whole world. This paper is an attempt to do just that. The code for this paper can be found on our Github page: https://github.com/trialanderror123/APSAPT.

Our main contributions through this paper are as follows:

1) We designed and implemented an end-to-end automated political tweets pipeline which requires just some basic configuration from the user. It handles all the sentiment analysis steps from data collection to visualization. The prediction is tested on 2 state-of-the-art deep learning networks: BERT (Bidirectional Encoder Representations from Transformers) and LSTM (Long Short-Term Memory). It reached an accuracy of 73.73% on the tested dataset.
2) A novel method to enhance the credibility of tweets based on their metadata like follower and like count which provides more accurate results.
3) A novel method to include polarity detection of tweets to make hand-labelling of data easier.

2375-9259/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDMW53433.2021.00022

4) The collation of a gold standard dataset of 2000 manually labelled tweets from 23 countries around the world and their opinion on whether they are Pro-Trump or Pro-Biden for the 2020 US Presidential Elections.

After the Introduction, the paper moves on to the Related Research section where more insights and general knowledge about our research areas are explained. Then in Methodology, we use our own use case of determining the preference (Trump or Biden) of crucial nations around the world on the US 2020 Presidential Elections to walkthrough the usage of our pipeline. Next, in Experiments and Discussion we provide the metrics and evaluate the prediction results of our pipeline for our use case. Lastly, in Conclusion and Further Work we sum up our contributions and list down further improvements to our pipeline.

II. Related Work

There have been multiple advances in the realm of NLP which have progressed data mining and analysis of tweets in social media. One major proponent is a new mode for RNNs (Recursive Neural Networks) to read text such that even the context is understood as popularized by LSTM [10] networks. The model architecture contains loops and gates which control the flow of information from one layer to another. The training goal is to decide what parts of the text are most important for a clear understanding of the text as a whole. This was the first attempt at featuring long-term memory in networks for better contextual cognizance of the text. The effectiveness of such a network was proven by a research that used two independent bidirectional LSTM and GRU (Gated Recurrent Unit) layers to perform sentiment analysis on three Twitter datasets [31]. Similar research was carried out to predict the degree of intensity for emotion in online corpora by using a stacked ensemble method utilizing LSTMs [12]. Another equally pertinent paradigm shift was brought in by the concept of transfer learning. It allowed AI practitioners to take a large pre-trained model trained on billions of generic text corpus taken from the internet and then fine-tune the model by training it on a smaller dataset that is more relevant to the final use case [13]. A method that achieved state-of-the-art results using transfer learning at its foundation is Universal Language Model Fine-tuning (ULMFiT) [14]. It works by first training the encoder part of an encoder-decoder transformer on a large unlabelled corpus of text which perpetuates general information about the grammar and sentence structure of the language being trained on. Later the encoder is fitted with a decoder trained on the down-stream task which ultimately creates the classifier for the final use case. These advances culminated in the invention of the BERT [15] model which reads context from both left and right sides of the text. There are even special BERT models like TweetBERT [16] which is pre-trained on 540 million English tweets which stand as popular options for social media analysis down-stream tasks. Our paper utilizes BERT for the pipeline's predictions so more information shall be provided in the Methodology section.

A. Political Tweets Pipeline

Big data analysis of tweets has always been a major component in political propaganda to size up the popularity of political candidates battling for supremacy. Hence, there have been a plethora of experiments done to improve classification results of tweet-like text corpora. For example, a research done on Italian tweets [17] came up with a few important contributions on best practices to follow when working with tweet-like sentence structures. They conjectured that emoticons are best handled by turning them into plain text rather than their unicode counterpart in order to take advantage of the fact that most large pre-trained language models are trained on plain text corpora. Moreover, they concurred that in order to handle tweets that are in multiple languages, it is far more beneficial to pre-train the model on plain text from those languages rather than tweets because plain text corpora are usually more in number than only-tweets corpora thus enabling better knowledge distillation [18] in the downstream task. This is the reason why we chose to use a standard BERT model instead of the more specific TweetBERT.

The idea of a pipeline which automates all the processes from data gathering to classification for an NLP task is a fairly novel concept with few successful implementations throughout history. [19] One of which uses a conglomeration of REST APIs and the Stanford NLP package to provide real-time trend analysis of certain words and brands. Another example [20] is a large-scale attempt to model public relations between multiple countries in the world map based on certain macro-economic sentiments. The overall global sentiment of a country is measured as a ratio between positive and negative tweets from other countries to the target country [21]. The success of such research works also shows credence towards our paper that it is indeed possible to chart out important political biases by analyzing Twitter feed data for the countries in question.

B. Polarity Detection

Polarity detection is an important prerequisite for opinion mining which deals with classifying certain ideas in unstructured textual data depending on the emotion it is trying to evoke [22]. It is often used by businesses to determine how customers feel about their brand and products by analyzing their sentiments on social media and product reviews [23]. Polarity detection usually consists of two macro-steps: first, to determine if the text is subjective or objective in nature. Next, if subjective, to determine if the text is positive or negative. Over the years, practitioners have come up with various techniques involving both rule-based and ML methodologies.

A popular rule-based approach is to refer to a comprehensive dictionary of positive and negative words that have ratings/intensities attached to each word depending on how strong of a word it is (ie - "love" is rated higher than "like"). Some such famous word nets include SentiWordNet [24], WordNet Affect [25], and Q-WordNet [26]. These word nets are utilized by running feature selection algorithms on the text which use n-grams or POS (Part Of Speech) ideas to

select the most important words in the text which lead to the highest information gain in the model. Next, the total positive or negative words can be tallied to provide a rough estimate of the overall sentiment [27]. The invention of such dictionaries led to a more robust framework of English grammar rules and lexicography called SentiStrength [28]. It uses pragmatic rules such as booster words like "very" and "fully" would multiply the sentiment of the next word and repeated letters in an adjective would intensify the emotion of that word (eg - "happppppy" has a more positive sentiment than "happy"). However, such rule-based models were deemed to be only good at predicting the sentiment for small texts spanning a few sentences. When used to get the big picture of a whole paragraph, these algorithms were seen to be ineffective.

Hence, we saw a shift towards more ML-centric methods of which a popular example was Senti4SD [29]. It used 4000 gold-standard human-annotated samples from busy sites such as Stack Overflow to train a Naive Bayes classifier which segregated the posts into 3 bins: positive, negative, and objective. It was able to improve the baselines set by SentiStrength by 19% to 25%. Another research followed an approach incorporating multi-level fine-scaled sentiment sensing with ambivalence handling to detect objectivity [30]. Yet another research targeted neutrality detection by building consensus vote models on online reviews [31]. The standard these days follows a mix of rule-based and ML-centric approaches which seem to outperform the past methodologies. One example is SenticNet [32] which we have used for our pipeline, more information will be provided in the Methodology section.

III. Methodology

This section will explain how we built the whole pipeline from start to end with reference to the dataset we scraped about different countries' opinions on Twitter about the 2020 US Presidential Elections. The overall design of our pipeline can be seen in Fig 1. This section may be used as a tutorial on how to use the pipeline for different datasets based on any other use case.

A. Data Mining and Cleaning

Our objective was to determine whether a country is Pro-Trump or Pro-Biden depending on the sentiments of their tweets on the topic of 2020 US Presidential Elections. With this in mind, we scraped over 100,000 tweets from 23 countries that have vested interest in the elections. The pipeline requires an array of all the relevant countries and their respective geographical city centres because the pipeline uses SNScrape [33] to scrape all the tweets containing a certain string within a predetermined radius from the city centre. The radius needs to be manually adjusted depending on the size of the country. The 40 hashtags used to scrape the tweets are mentioned in Table 1.

Our pipeline also allows the user to input the period of time between which the tweets can be scraped from. For our case, we scraped from 1st November 2020 to 30 November 2020 because that was the election month which garnered the most worldwide discussions. Even the language of the scraped tweets can be restricted to a few, we only chose English. The following metadata was kept along with each tweet: username, content, date, country, reply count, retweet count, like count. The final 3 factors would be used later during the weighted averaging of the results.

After the initial scraping, we realized that most of the tweets were irrelevant because the hashtags spanned longer than the actual message in the tweet or there were duplicate tweets from the same user. Python scripts were used to remove these extraneous tweets. After this step the total tweet count was reduced to 38,364. A script was used to remove the Neutral tweets because we found out that the model performs better using subjective tweets. After these pre-processing steps which are all automated into our pipeline, the final count for the gold-standard dataset is 27,146 tweets. Some examples of these tweets are as follows: "The polls tell us Biden is winning by a landslide, but no one turns out for his events and apart from "he's not Trump" no one in his camp can come up with a reason to vote for him. Tens of thousands turn out for Trump every day" and "Joe Biden lies about his corruption?? He said he never spoke to his son about his overseas business dealings?? He said there was not a shred of evidence of a quid pro quo (despite announcing it on tape)??". We hand-annotated 2000 of these into 3 categories: Pro-Trump, Pro-Biden, and Neutral. The inter-annotator score as measured by the Fleiss' kappa score is 0.9370 which is above the standard 0.80 thus making these annotations trustworthy. The user may choose to not run any of the above steps if it does not make sense for their particular dataset. Finally, the dataset was tokenized using an external python library called sentencepiece [34] and word embeddings were created out of them.

B. Utilizing Deep Learning Models

To generate predictions from the dataset and determine which Deep Learning model works best for our pipeline, we utilized 2 state-of-the-art NN models: LSTM and BERT. After experimentation, it was concluded that BERT outperformed LSTM in all the test metrics so BERT was chosen as the model best suited for our pipeline. Hence, this section will only touch upon some of the main details of the LSTM model while mainly focusing on the BERT model instead. The final metrics for both models can be found in the Experiments and Discussion section.

The LSTM network we used has a sigmoid layer coupled with a pointwise multiplicator which outputs a float between 1 and 0. This layer acts as a "gate" which controls the flow of information to each layer essentially allowing the network to add or remove information according to what it deems to be relevant. The model was first pre-trained on WikiText-103 [35] then fine-tuned on our labelled dataset. ULMFiT techniques were used during training such as gradual unfreezing of layers and dynamic learning rate scaling [14].

For the BERT training, we used the BERT_Base model which contains 12 layers (transformer blocks), 12 attention heads, and 110 million parameters [15]. Unlike classic NLP

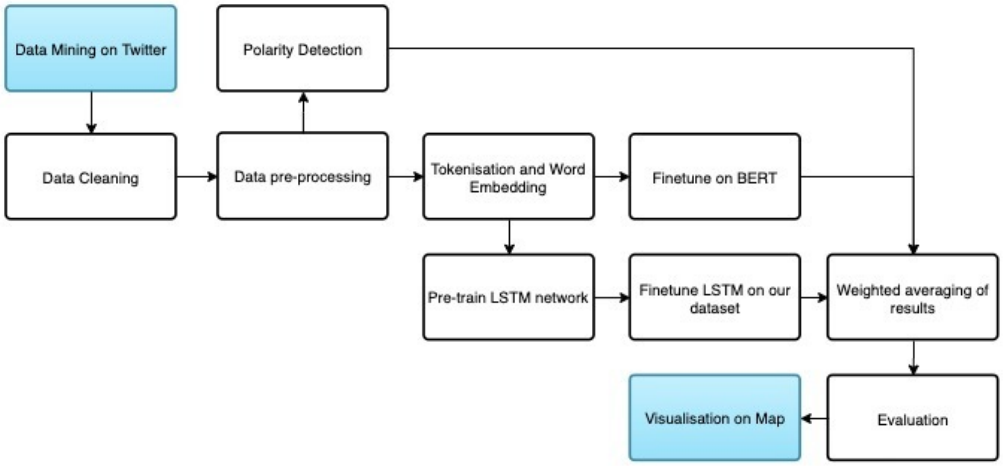

Fig. 1. Overview of our pipeline

TABLE I
PHRASES AND HASHTAGS USED FOR DATA MINING

Mentions of Trump	Pro-Trump	Anti-Trump	Mentions of Biden	Pro-Biden	Anti-Biden
#Trump	#MAGA	#VoteTrumpOut	#Biden	#VoteBiden	Sleepy Joe
#trump	#PresidentTrump	#DumpTrump	#biden	VoteBlue	#SleepyJoe
#Trump2020	VoteRed	#TrumpIsPathetic	#Biden2020	#VoteBlueToSaveAmerica	HidenBiden
#DonaldTrump	#MakeAmericaGreatAgain	#TrumpCorruption	Joe Biden	#BlueWave2020	#CreepyJoeBiden
DonaldJTrump	#TeamTrump	#VoteHimOut	#JoeBiden	#TeamBiden	#NeverBiden
Donald Trump	#VoteTrump	#YoureFiredTrump	Biden	#JoeMentum	#BidenUkraineScandal
Trump	#DrainTheSwamp	#TrumpHasToGo			#HunterBiden
		#MyPresident			

models of reading text, BERT reads the text as a whole such that both the contexts to the left and right of each word are understood by the model. Hence, a max sentence length needs to be configured which was kept at 100 for our dataset because tweets would never cross that threshold. The user may change this parameter in the config file of our pipeline according to their utilization.

Our BERT model makes use of 2 pre-training techniques when creating the language model for the dataset. First it masks 15% of train data using a special "MASK" token which it then tries to predict during the training phase thereby learning the contextual positioning of that word in the text. The second technique is using positional embeddings for each word so that the model does not forget the placement of a word in the overall text. This is used to predict the next word in the sentence [36]. The architecture of the BERT model is given in Fig 2.

The dataset of 2000 labelled data is split into train-validation-test by the ratio 80:15:5. The batch size is 8 and the model trained for 5 epochs. The hyperparameters are as follows: Weight Decay Rate = 0.01 and Learning Rate = 3e-5. The Adam optimizer was used to reach convergence at a faster rate. A Google Colaboratory notebook with CUDA enabled was used for this experiment. The above parameters and more can be tuned from the config.py file in our pipeline however the default is set as above because it works well on most datasets as determined from literature review of many deep learning research papers.

C. Polarity Detection

Polarity detection was done using the SenticNet 6 API. It outputs 1 of 3 labels: positive, negative, or neutral. An extra "error" label is output if the API cannot understand the text. The API utilizes a top-down symbolic (semantic network or logic model) approach to decode meaning from the text as well as a bottom-up sub-symbolic approach using biLSTM and BERT deep learning models to learn syntactic patterns from the data [37]. First the API tries to break down the text into primitives which are just conceptual building blocks of a language. Eg - "pasta" and "pizza" would be classified as FOOD [38]. By doing this pre-processing step, the text is broken down to its most basic meaning and each primitive is given a polarity based on dependency rules and the total polarity is just a summation of these basal polarities [39]. If sentic patterns cannot be extracted using this method, the API uses the BERT architecture to obtain sentential contextual embeddings of the text. The model is trained on the all the concepts of the verb-noun and Adjective-noun forms that are

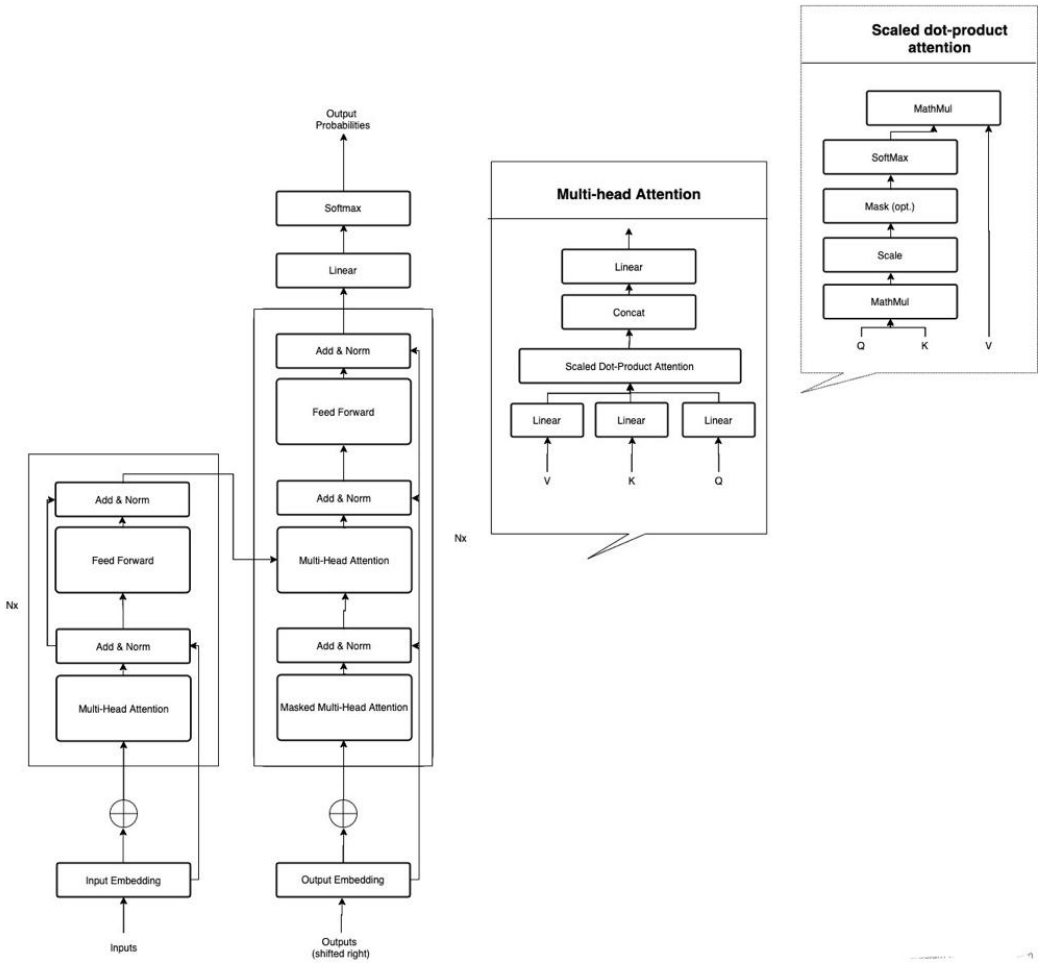

Fig. 2. Architecture of BERT

present in ConceptNet5 [40]. A biLSTM [41] has also been experimented with to provide the aforementioned contextual embeddings. These embeddings are then used to generate the primitives which help to define the logical meaning of the text from which polarity can be extracted.

D. Weighted Averaging Results

The objective of the pipeline is to utilize existing sentiment-analysis models by adapting them for political social-media tweets. An intuitive way to gauge the importance of these tweets is through tweet-metadata. The likes, retweets and replies were the engagement metrics considered while evaluating the weighted sentiment. Additionally, the polarity score from Sentic API was considered, with the rationale that more polarized tweets (and its engagement) indicate stronger support for a particular viewpoint.

$$wLikes = 1 + log(2 * likes)$$

$$wRetweets = 1 + log(2 * retweets)$$

$$wReposts = 1 + log(2 * reposts)$$

$$wSentiment = wLikes{*}wRetweets{*}wReplies{*}sPolarity^2$$

The 'wSentiment' score is intended to be a representation of the influence and strength in beliefs of the tweet. These scores are an indicator of how credible a certain tweet is because it follows the rationale that a tweet that has been retweeted a lot or has harbored many likes and is from an account with multiple followers should have more weight in swaying the predictions than just a generic tweet. These scores are later put to use during the visualization phase of our pipeline.

E. Data Visualization

In order to better visualize the prediction results, our pipeline generates 2 types of graphs which provide more insight on the data:

1) Word clouds: One word cloud per class is created which would contain the highest word frequencies present in

that class. The relative size of each word shows its frequency in the text corpus.

2) World map: The weighted sentiment score for each country is taken for all the classes and the scores are normalized proportionately between 0 and 1. The dominating value among the classes is taken and is visualized in a world choropleth map.

IV. Experiments and Discussions

A. Classification Results

The predictions of each model were evaluated based on 4 metrics: precision, recall, test accuracy and the F1 score [42].

$$F1 = 2 * (\frac{precision * recall}{precision + recall}) = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FP}$$

Where, TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

The results of the prediction task can be seen in Table 2. As mentioned in Methodology, BERT outperforms LSTM in all metrics with a F1 score of 74.49%. One reason that the LSTM model did not give good scores might be because the model was overfitting on the data as was implied by the extremely high train accuracies (95+%). We tried to solve this by augmenting the data using the Synonym Replacement and the Random Insertion techniques proposed by the EDA [43] paper. However, the results only improved minutely. This shows that using LSTM would require much more than just 2000 labelled instances and is not suitable for small to medium-sized datasets. Since we wanted our pipeline to be inclusive to as many real-life datasets as possible, we decided to use BERT for the final iteration of our pipeline.

TABLE II
Classification Results

Model Name	F1 Score	Accuracy	Precision	Recall
LSTM	0.56300	0.56300	0.68921	0.46454
BERT	0.74490	0.73737	0.75258	0.73737

B. Polarity Detection Results

The SenticNet 6 API was used to extract the polarities of all the tweets in the labelled dataset. The total count of each polarity is given in Table 3. The API only produces an error state 4.85% of times, proving its strong reliability. The polarity detection module is implemented in our pipeline as a way of providing extra information or metadata to the user which can provide more insights about their data. A more practical use of this data can be to aid in the process of automatic labelling of the user's data. We hand labelled the US Presidential Elections data to avoid any ambiguity but it is possible to first find out whether the tweet is directed to either Trump or Biden by using basic string searching techniques. Next, the polarity detection can be run to determine the tweet's sentiment towards the candidate in question (eg - a tweet containing hashtags invoking "Trump" which is positive in polarity might be labelled as Pro-Trump). Of course, this method is not 100% accurate but it can provide a suitable foundation on which to begin hand labelling if the user's data contains too much data to manually label from scratch.

TABLE III
Polarity Detection Counts

Positive	Negative	Neutral	Error
946	591	280	97

C. Visualization Results

Our pipeline provides word cloud visualizations of all the unique classes of the user's dataset. The ones from our dataset can be viewed in Figs 3-5. It is interesting to see that "trump" appeared around the same frequency of times as "biden" in the Pro-Biden tweets implying that most tweets that were labelled as advantageous for Biden were labelled so because they were demeaning Trump. Most of the hashtags used to scrape the relevant tweets can be seen in the word clouds as well.

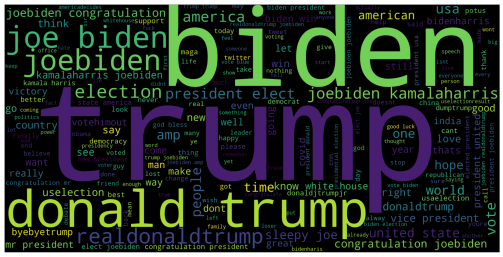

Fig. 3. General word cloud

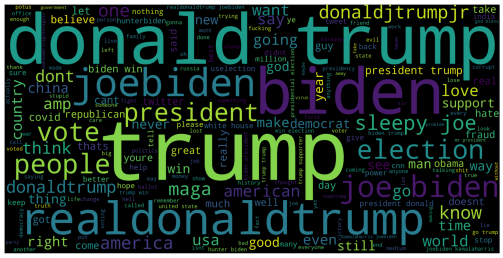

Fig. 4. Pro-Trump word cloud

Furthermore, since our pipeline is geared towards analyzing political tweets, we are visualizing the sentiments of all the nations related to the political event on a world map as shown in Fig 6. The colors are inferred from the weighted sentiment analysis results. In this map, the color red represents Pro-Trump nations and color blue represents Pro-Biden nations. This map is created using the plotly and pycountry [44] packages in Python. It can be observed that Hong Kong,

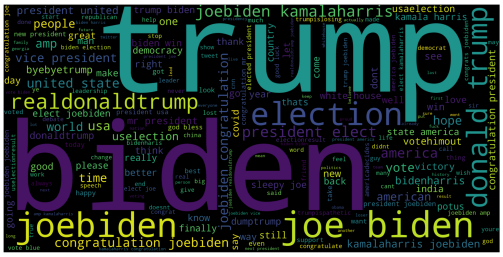

Fig. 5. Pro-Biden word cloud

Ukraine, Saudi Arabia, Nigeria, and Japan are the only Pro-Trump countries with the rest of the countries being Pro-Biden. So from these results it can be concluded that the world in general prefers Biden over Trump. However, It should be noted that Twitter likely has a Biden bias as it primarily consists of younger users who are traditionally more liberal. Thus, these visualizations provide interesting inferences to the user of our pipeline based on their data.

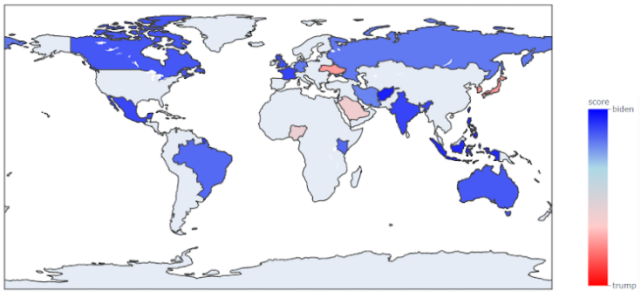

Fig. 6. Sentiments of key nations visualized on a map

V. Conclusion and Further Work

In this paper, we explained the creation and usage of our political pipeline which uses NLP techniques to gather insights about certain political events and how key nations around the world feel about it. This pipeline is engineered with the prime objective that the user should only be required to handle certain aspects of the whole process, the rest being automated by the pipeline. The pipeline begins by scraping Twitter for tweets relevant to the use case to build a dataset. Then, these tweets are cleaned and pre-processed to make them ready for training the deep learning model. We used BERT for the predictions because it is a state-of-the-art network used for many NLP tasks today. The predictions are then weighted and averaged based on the tweet's like and follower count to determine the credibility of the tweet. Polarity detection is also done to provide the user with more metadata about the tweets. Finally, the predictions are segregated by country and visualized on a map with varying color shades to show the varying sentiments of each country. The whole process is tested with our personal use case which was to determine which countries in the world are Pro-Biden or Pro-Trump for the 2020 US Presidential Elections. 73.73% test accuracy was obtained which is a good indicator that our pipeline is indeed effective.

Our research, although ambitious, can have multiple avenues of improvement in the future. One would be to give the user the option to choose alternative deep learning or machine learning models other than BERT depending on their use case. Another would be to implement sarcasm detection which could improve prediction accuracy because Twitter as a social media is highly susceptible to sarcastic comments which might confuse a neural network if not trained with enough data. Finally, implementing another visualization graph which shows how political trends changed over time might output useful information (eg - US Presidential Elections from 1990 to 2020).

References

[1] E. Cambria, S. Poria, A. Gelbukh and M. Thelwall, "Sentiment Analysis Is a Big Suitcase," Affective Computing and Sentiment Analysis, pp. 74-80, 2017.
[2] E. Cambria, H. Wang, and B. White, "Guest Editorial: Big Social Data Analysis," Knowledge-Based Systems, pp. 1-2, 2014.
[3] E. Cambria, M. Grassi, A. Hussain, and C. Havasi, "Sentic computing for social media marketing," Multimedia Tools and Applications, vol. 59, no. 2, pp. 557-577, 2012.
[4] E. D. Hersh, Hacking the Electorate: How Campaigns Perceive Voters. Cambridge: Cambridge University Press, 2015.
[5] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau, "Sentiment analysis of twitter data," in Proceedings of the workshop on language in social media (LSM 2011), 2011, pp. 30-38.
[6] E. Xia, H. Yue, and H. Liu, "Tweet Sentiment Analysis of the 2020 US Presidential Election," in Companion Proceedings of the Web Conference 2021, 2021, pp. 367-371.
[7] W. Budiharto and M. Meiliana, "Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis," Journal of Big data, vol. 5, no. 1, pp. 1-10, 2018.
[8] A. Khatua, E. Cambria, A. Khatua, and I. Chaturvedi, "Let's chat about brexit! a politically-sensitive dialog system based on twitter data," in 2017 IEEE International Conference on Data Mining Workshops (ICDMW), 2017: IEEE, pp. 393-398.
[9] A. Khatua, A. Khatua, and E. Cambria, "Predicting political sentiments of voters from Twitter in multi-party contexts," Applied Soft Computing, vol. 97, p. 106743, 2020.
[10] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM–a tutorial into Long Short-Term Memory Recurrent Neural Networks," arXiv preprint arXiv:1909.09586, 2019.
[11] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," Future Generation Computer Systems, vol. 115, pp. 279-294, 2021.
[12] M. S. Akhtar, A. Ekbal, and E. Cambria, "How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]," IEEE Computational Intelligence Magazine, vol. 15, no. 1, pp. 64-75, 2020.
[13] F. Zhuang, "A comprehensive survey on transfer learning," Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, 2020.
[14] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," arXiv preprint arXiv:1801.06146, 2018.
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[16] M. M. A. Qudar and V. Mago, "TweetBERT: A Pretrained Language Representation Model for Twitter Text Analysis," arXiv preprint arXiv:2010.11091, 2020.
[17] M. Pota, M. Ventura, R. Catelli, and M. Esposito, "An effective BERT-based pipeline for Twitter sentiment analysis: a case study in Italian," Sensors, vol. 21, no. 1, p. 133, 2021.
[18] X. Jiao, "Tinybert: Distilling bert for natural language understanding," arXiv preprint arXiv:1909.10351, 2019.

[19] H. Jain, "A Web Based Application for Sentiment Analysis," International Journal of Education and Management Engineering, vol. 1, pp. 25-35, 2017.
[20] N. Chambers, "Identifying political sentiment between nation states with social media," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 65-75.
[21] N. Chambers, "Identifying political sentiment between nation states with social media," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 65-75.
[22] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Comput. Linguist, vol. 35, no. 2, pp. 311-312, 2009.
[23] "What is opinion mining and why is it essential?," MonkeyLearn Blog, 16-Sep-2020. [Online]. Available: https://monkeylearn.com/blog/opinion-mining/. [Accessed: 24-Aug-2021].
[24] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in Lrec, 2010, vol. 10, no. 2010, pp. 2200-2204.
[25] C. Strapparava and A. Valitutti, "Wordnet affect: an affective extension of wordnet," in Lrec, 2004, vol. 4, no. 1083-1086: Lisbon, p. 40.
[26] I. S. Vicente, R. Agerri, and G. Rigau, "Q-wordnet ppv: Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages," arXiv preprint arXiv:1702.01711, 2017.
[27] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," ACM transactions on information systems (TOIS), vol. 26, no. 3, pp. 1-34, 2008.
[28] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," Journal of the American society for information science and technology, vol. 61, no. 12, pp. 2544-2558, 2010.
[29] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," Empirical Software Engineering, vol. 23, no. 3, pp. 1352-1382, 2018.
[30] Z. Wang, S.-B. Ho, and E. Cambria, "Multi-level fine-scaled sentiment sensing with ambivalence handling," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 28, no. 04, pp. 683-697, 2020.
[31] A. Valdivia, M. V. Luzón, E. Cambria, and F. Herrera, "Consensus vote models for detecting and filtering neutrality in sentiment analysis," Information Fusion, vol. 44, pp. 126-135, 2018.
[32] Sentic.net. 2021. SenticNet. [online] Available at: ¡https://sentic.net/¿ [Accessed 24 August 2021].
[33] "GitHub - JustAnotherArchivist/snscrape: A social networking service scraper in Python", GitHub, 2021. [online] Available at: ¡https://github.com/JustAnotherArchivist/snscrape/¿ [Accessed 24 August 2021]
[34] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," arXiv preprint arXiv:1808.06226, 2018.
[35] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," arXiv preprint arXiv:1609.07843, 2016.
[36] R. Mohd Sanad Zaki , "What is BERT: BERT For Text Classification," Analytics Vidhya, 14-Jun-2020. [Online]. Available: ¡https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/#h2_4/¿ [Accessed: 24-Aug-2021].
[37] E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, "SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis," in Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 105-114.
[38] E. Cambria, T. Mazzocco, A. Hussain, and C. Eckl, "Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space," in International Symposium on Neural Networks, 2011: Springer, pp. 601-610.
[39] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in Proceedings of the AAAI conference on artificial intelligence, 2018, vol. 32, no. 1.
[40] R. Speer and C. Havasi, "ConceptNet 5: A large semantic network for relational knowledge," in The People's Web Meets NLP: Springer, 2013, pp. 161-176.
[41] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in Thirty-second AAAI conference on artificial intelligence, 2018.
[42] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in European conference on information retrieval, 2005: Springer, pp. 345-359.
[43] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," arXiv preprint arXiv:1901.11196, 2019.
[44] "pycountry," PyPI. [Online]. Available: https://pypi.org/project/pycountry/. [Accessed: 24-Aug-2021].