AI-Assisted Tool for Marketing Exploration

Group 3

Group Members:

First name	Last Name	Student number
Ashna	**Prasannan**	**C0786430**
Jiya	**Peter**	**C0789655**
Najna	**Nazeer**	**C0793127**
Rincy	**Jose**	**C0790298**
Duc Anh	**Trinh**	**C0791920**

Submission date: ***2021 December 17***

Contents

Capstone Project

Abstract

As of January 2021, Twitter had approximately 6.45 million active users in Canada, accounting for almost 17% its population. Twitter is a hugely potential source of information for marketing exploration. In this project, we generate a tweet sentiment classifier to provide customers the opinion of Twitter community about a product or a service of interest. We firstly take advantage of manually annotated tweets to train the classifier. The dataset was preprocessed, encoded, and transformed using various Python libraries before feeding to deep learning models. The best model will be chosen as the final classifier. In our system, the information that customers are interested in, will be collected from Twitter in real-time manner. The retrieved tweets are processed and classified using the established classifier to extract useful information that presents back to the customer in graphs or statistic numbers. We finally package the procedures into a web app using a powerful web framework, composed of Django and Django REST API, which benefits both customer and administrator to handle information at ease. The source code is managed with GitHub repository and available online in PythonAnywhere clouding.

1. Introduction

Twitter has become an important social media platform in many countries in the world. Over one in five adults in USA actively use Twitter in daily basis (Hughes A. and Wojcik S., 2019). In Canada, the number is even more impressive, 15 million Canadians 'tweet' monthly, accounting for 49.7% of Canada's online population (Slater M., 2018). The fact indicates the important of Twitter as information source for a customer service. In fact, the tweets are exploited to provide useful reference in public perception or mood, especially about public health (Carpenter, J. et al., 2020; Tavoschi, L. et al., 2020).

When exploring sentiment analysis to retrieve the public perceptions, two approaches are lexicon-based, and machine learning (ML) based (Parveez S. & IriondoSentiment R., 2020). Two most prominent lexical tools are TextBlob and VADER (Pope L., 2020). This lexical or rule-based method is popular, easy-to-apply and convenient, but the prediction accuracy may not be consistent with different specific domains of knowledge (Kaur H., & Mangat V., 2027). For example, the word "disease" is usually marked as negative in general dialogue but should be neutral in a scientific discussion. The second approach aims to overcome that challenge. The ML-based classification heavily depends on dataset for training. The more target texts are closed or same in corpus with the train dataset, the better performance of the model is.

Many methods have been developed to process the tweets to compatible with machine learning or deep learning algorithms to maximize the utilization of information from tweets. In principle, the tweet needs to process to remove unwanted factors (hyper-linked texts for example), tokenized and transformed into numeric vectors that can be read by machine. A powerful method to deal with text is that focuses on the sequence of words in text - string-of-words method. The order of words is one of the most important in sentence meaning. Hence, a deep learning technique called Recurrent Neural Network (RNN) armed with Long-Short Term Memory (LSTM) has emerged as the most popular method for text analysis. LSTM is delighted with its ability to learn, remember, and provide prediction from sequences (Tyagi, V., et al. 2020). Before the data is fed to the model, the text is also required to be

tokenized and transformed to numeric vectors that represent their corresponding positions in the corpus (embedding step).

To collect tweets, several tools are developed to connect API provided by Twitter. Twitter provides limit rate based on types of customers. The most popular tool written in Python is Tweepy. Although there are several limitations on rate and number of downloads, Twitter API-based method provides full feature of tweets, including geographic, location, and some other tools can pull the tweets down without using API. They crap all tweets using some settings of keywords or time tunning. However, the retrieved tweets may lack some information (for example, longitude and latitude) compared the API-based method.

Among various Python-based web frameworks, Django enables rapid development of secure and maintainable websites, helping the developer focus on developing application instead of repeating coding from the scratch. Django is powerful to build both web app and API backend that can integrate well with the python-based machine learning models.

We here aim to build a machine-based tweet sentiment classifier that can provide useful information from public tweets in real-time and robust manner. The classifier is embedded in Django framework powered with Restful API, easy to manage and scale up.

2. Methods

2.1. Tools and program languages

In this project, most tools are Python-based libraries. Except for some html codes were used to build front-end web framework, all steps were written in Python. In details, data preprocessing and modeling were conducted with Pandas, NumPy, Regular Expression, NLTK, TextBlob, Scikit-learn. The RNN model was built using TensorFlow and Keras on top. Visualization was made with Matplotlib. The models were saved and loaded with Joblib. Tweets were collected using Tweepy. Web deployment was built on Django and Django REST Framework.

2.2. Datasets and text pre-processing

To train a model for tweet sentiment evaluation, we need pre-labelled datasets. There are two types of pre-labelled datasets: automated label and manual label. For the former, the judgement is based on machine algorithms to calculate the distant values to make polarity of phrases. Hence it can be very large but contain much noise. In contrast, the latter is classified by human, more human but small in amount. Here we combine 4 different manual –labelled datasets to ensure big enough for training data.

- Semeval: The SemEval [27] corpus is formed by 5232 positive tweets and 2067 negative tweets annotated by human evaluators using the crowdsourcing platform Amazon Mechanical Turk.
- 6HumanCoded. The 6HumanCoded dataset is a collection of 1340 positive and 949 negative tweets scored according to positive and negative numeric scores by six human evaluators.
- Sanders. The Sanders dataset consists of 570 positive and 654 negative tweets evaluated by a single human annotator.

- Twitter US Airline Sent: Twitter data was scraped from February of 2015 and contributors were asked to classify positive, negative, and neutral tweets.

We join 4 datasets to a unique data set containing 27000 pre-labeled tweets for model training. The sentiment distribution is quite balanced, 0 for negative, 1 for neutral and 2 for positive tweets.

Pre-processing includes removing URLs, html tags, hashtags, mentions and remove anything not a letter. We also remove punctuations, non-alphabet characters, then tokenize them. To feed to an RNN model, we need to vectorize tweet texts, converting words into numeric representation. GloVe or Global Vectors for Word Representation is an unsupervised learning algorithm for obtaining vector representations for words, this model is trained on 2 billion tweets, which contains 27 billion tokens, 1.2 million vocabularies. Here we found 39676 unique tokens in our dataset.

2.3. Modeling

Recurrent Neural Networks (RNN) are good at processing sequence data for predictions. Therefore, they are extremely useful for deep learning applications like speech recognition, speech synthesis, natural language understanding, etc.

Three are three main types of RNNs: SimpleRNN, Long-Short Term Memories (LSTM), and Gated Recurrent Units (GRU). SimpleRNNs are good for processing sequence data for predictions but suffers from short-term memory. LSTM's and GRU's were created as a method to mitigate short-term memory using mechanisms called gates. In the build model function, we add dropout for overfitting prevention and compare among models based on valuation accuracy.

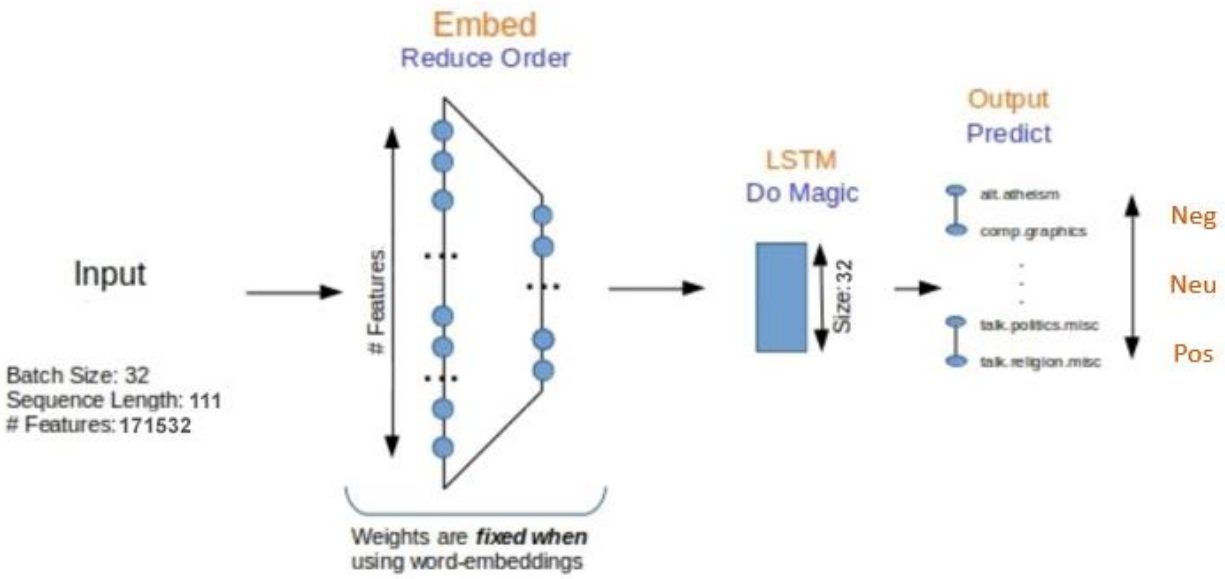


Figure 1. A simple RNN model

For each model, we set 20 epochs, including the Early Stopping condition, which is based on validation accuracy. The we collect the report for major metrics and visualize the loss and accuracy over epochs in both train and test sets. The best model will be saved for later prediction work.

2.4. Tweet Collection and Sentiment Analysis

We use Tweepy library to get tweets. The Twitter API belongs to standard track and the parameters, including tokens and authentic keys are set up. We restricted the tweets in English and excluded retweets and replies in search.

Figure 2. Setting up Tweepy parameters for tweet retrieval.

Collected tweets are converted to a data frame using Pandas library and processed as described at 2.1. The prediction workflow includes the cleaning (remove stopwords, lemmatization, clear away mentions, url non-letter); Tokenizer, vectorization and padding sequence to ensure fit to the model. The sen_eval function will return the sentiment value (negative, neutral, or positive) with the probability.

To explore further about sentiment analysis, use Matplotlib, Wordclouds to visualize the contents of negative or positive tweets.

2.5. Django web framework and Restful API

We deployed the model to a Django web framework locally. The installing procedure of the virtual environment and Django was followed the instruction in the Django website. All dependencies were installed into the virtual environment using pip package.

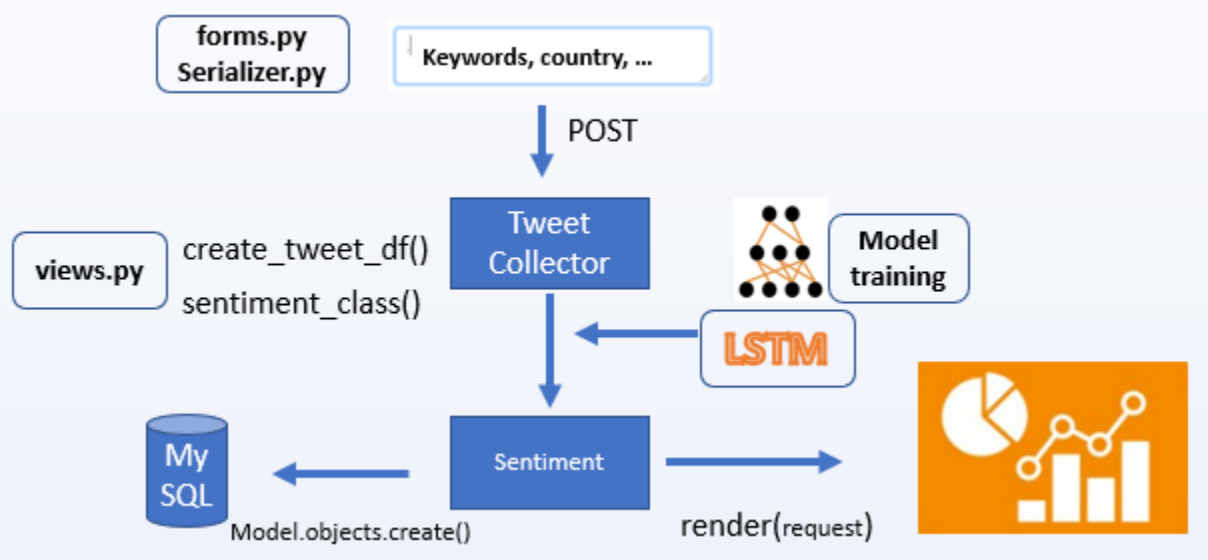


Figure 3. Django web framework integration.

We created an app named “homepage” to implement the classifier as illustrated in Figure 10. In brief, an html file (demo.html) contains a form that receives request from customers and make a POST request. The function demo() in views.py will handle the request and call series of other functions, which are store in functions.py file. They collect tweets, process, and analyze sentiment. The result is stored in database and sent back to the demo.html to present to customers. All information interactions between html file and views.py are directed in urls.py.

The Restful API is built with Django REST Framework (DRF). We create a serializers.py that define the method to interact with database and convert to Json format. The endpoints are also defined in urls.py. The URL API classifier/ shows up list of content in database after serialization, including id of every record. This page also provides ideas about using specific methods (PUT, GET for examples) to hook with endpoints in order to manage database through the restful API.

2.6. Clouding deployment

After everything (cleaning, modeling, tweet collection, sentiment analysis etc..) run smoothly in Jupiter notebook, we order functions and commands in order to work with Django framework. The version control and source code management are ensured by git. We usually commit and push to GitHub repository weekly. For clouding deployment, from bash-console provided by PythonAnywhere, we clone the source code directly from GitHub. The subsequent update can be pulled from GitHub or edit directly using built-in editor of PythonAnywhere.

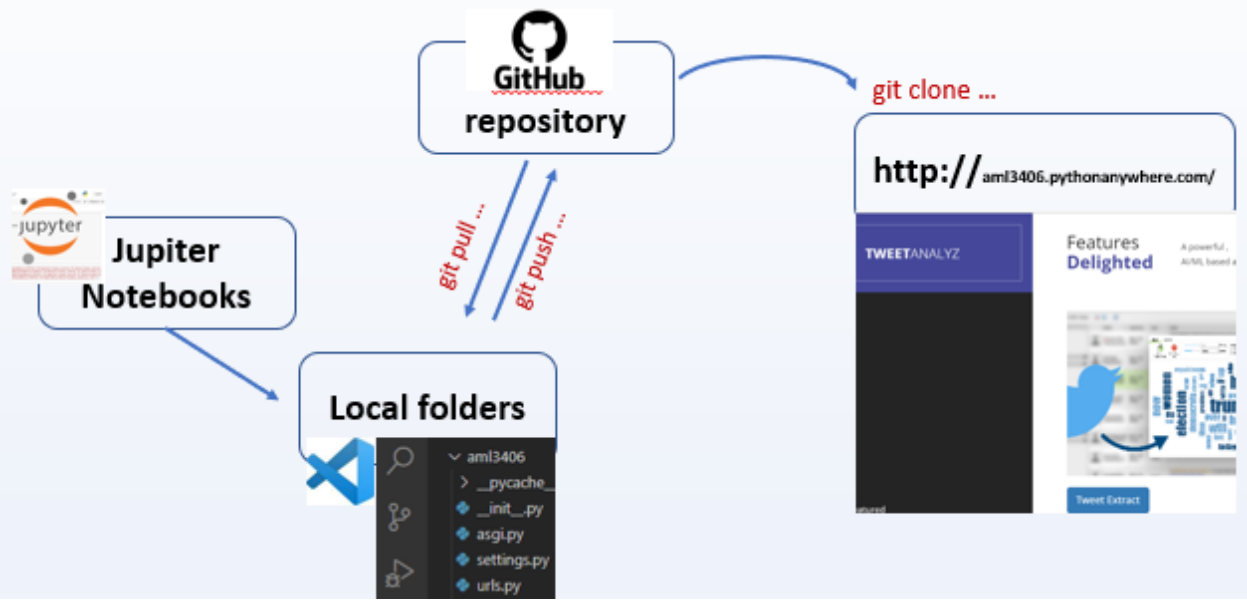


Figure 4. Source code management and clouding deployment.

3. Results

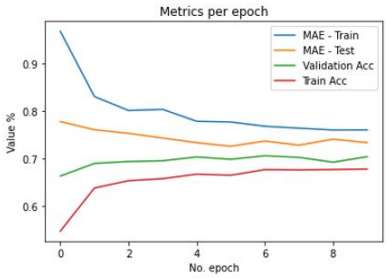
3.1. Tweet Classifier

	Text	Tweet_punct	clean_tweets	Tweet_tokenized
0	b"RT @rutoogenre: He's still smart now, during tmap golden bell challenge he's the only who got all the questions correct till the end / he g\xe2\...	bRT rutoogenre Hes still smart now during tmap golden bell challenge hes the only who got all the questions correct till the end he gxexxa	still smart now during golden bell challenge the only who got all the correct till the end he	still smart now during golden bell challenge the only who got all the correct till the end
1	b"@MrGee54 She's looking for Deion's son since she has history with JSU QBs..."	bMrGee Shes looking for Deions son since she has history with JSU QBs	looking for son since she history with	looking for son since she history with
2	b'@lnkslasher I have theory that bell is still alive, it\xe2\x80\x99s weird but hear me out, in the cod mobile 2nd anniversary\xe2\x80\xa6 https:...	blnkslasher I have theory that bell is still alive itxexxs weird but hear me out in the cod mobile nd anniversaryxexxa	I have theory that bell is still alive weird but hear me out in the cod mobile	have theory that bell still alive weird but hear out the cod mobile

Figure 5. Tweets undergo a series of cleaning steps.

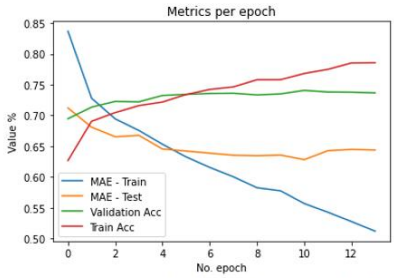
While there is a similarity in accuracy between LSTM and GRU models, SimpleRNN is significantly lower.

In general, all three models showed low biased outputs.



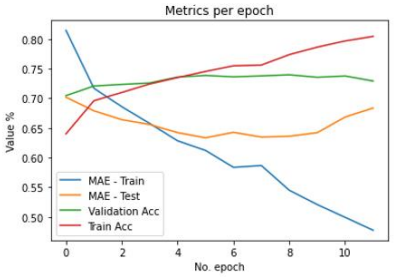
	precision	recall	f1-score	support
0	0.71	0.85	0.78	2203
1	0.70	0.64	0.67	2057
2	0.68	0.55	0.60	1171
accuracy			0.70	5431
macro avg	0.70	0.68	0.68	5431
weighted avg	0.70	0.70	0.70	5431

SimpleRNN
Accuracy: 0.70



	precision	recall	f1-score	support
0	0.79	0.80	0.80	2203
1	0.70	0.73	0.72	2057
2	0.68	0.62	0.65	1171
accuracy			0.74	5431
macro avg	0.73	0.72	0.72	5431
weighted avg	0.74	0.74	0.74	5431

LSTM
Accuracy: 0.74



	precision	recall	f1-score	support
0	0.74	0.87	0.80	2203
1	0.75	0.65	0.69	2057
2	0.67	0.61	0.64	1171
accuracy			0.73	5431
macro avg	0.72	0.71	0.71	5431
weighted avg	0.73	0.73	0.72	5431

GRU
Accuracy: 0.73

Figure 6. Comparison of performance among three RNN models.

However, epochs over the bigger overfitting – the gap between train/test wider. We think that the noises in dataset might be the cause. Therefore, we decided to clean more thoroughly the dataset using lemmatization and stemming. The better result of LSTM model below shows the lower overfitting level.

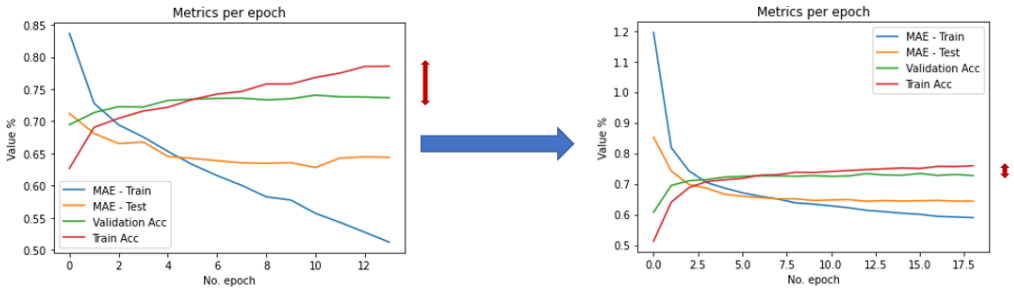


Figure 7. Cleaned dataset helps to reduce overfitting in RNN modeling.

3.2. Tweet Collection and Analysis

Using Tweepy, we could collect tweets with specific combinations of keywords. The collected tweet was stored in a data frame for the consequent analysis. Because of rate-limit policy of Twitter for a free account, we saved the privilege by only setting to collect maximum 50 tweets for one search. After applying the same cleaning steps as the dataset, we applied the previous classifier to label single tweet.

	Tweet_tokenized	Sent
0	still smart now during golden bell challenge the only who got all the correct till the end	Positive
1	looking for son since she history with	Neutral
2	have theory that bell still alive weird but hear out the cod mobile	Negative
3	Ugh ugh catch string but little bit She hold string again	Negative
4	true and every family show had Cockroach Skippy Family Screech	Positive

Figure 8. Tweet sentiment prediction.

Sentiment scores are collected and analyzed to generate statistical values such as sentiment distributions, pattern of words in a kind of sentiment tweets and the most popular words. Those values may reflect the typical characteristic of a business.

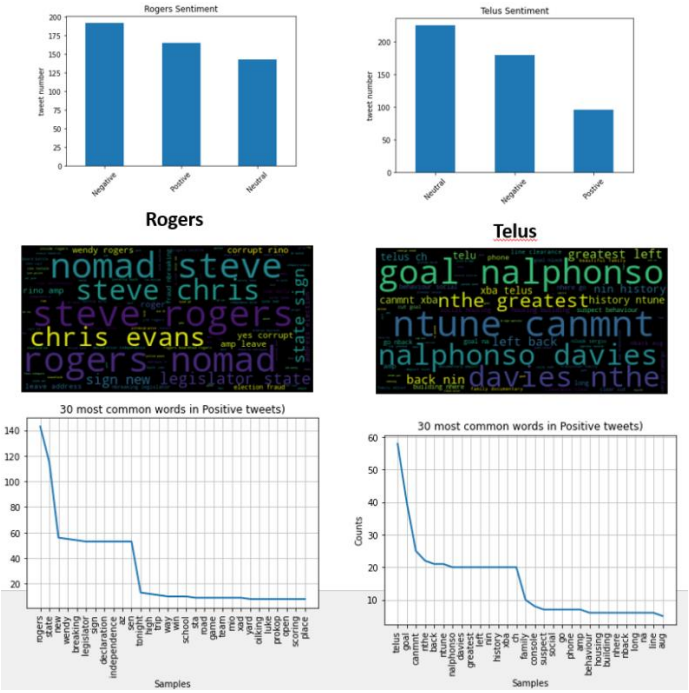


Figure 9. Insights of sentiment analysis for two business subjects Rogers and Telus mobile in Canada.

3.3. Web Framework Integration and Restful API

We build a web application for our classifier based on Django web framework, which was installed in a Python built-in virtual environment - venv.

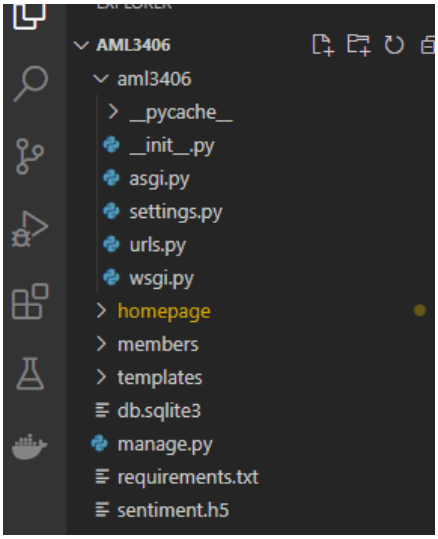


Figure 10. Root folder structure of the web application.

For front-end theme, we applied the latest stable version of Bootstrap, which allow the website structure to be lively, flexible, and adaptive to any screen size, on any platform. Html5 empowers the forms that they are validated by themselves with some specific kinds of data such as datetime or email.

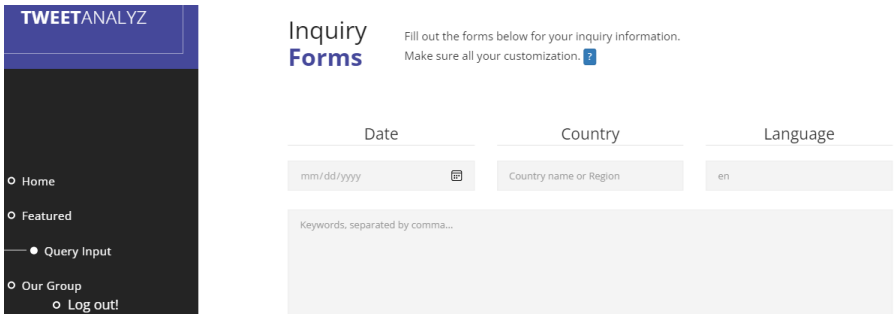


Figure 11. The interface of input forms.

3.4. Clouding Deployment

We use GitHub repository as a source code management. Weekly or bi-weekly sprint is committed pushed to git branches with the repository through a CLI from member pcs.

The code in GitHub is then cloned to PythonAnywhere bash console. In PythonAnywhere platform, we config settings.py to use MySQL and other parameters specific for the platform.

4. Conclusions and Future Work

In TweetAnalyz, we did train a Twitter content classifier based on a pre-labelled tweet dataset and applied to analyze the real-time content collected from Twitter. We built a simple but powerful web application that not only provide a service of automated sentiment analysis for customer, but also benefit the company with the ease of management.

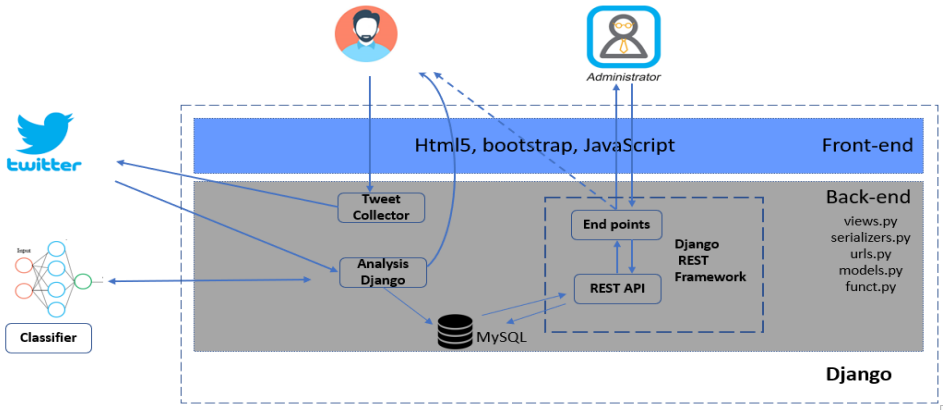


Figure 12. Schematic operation of TweetAnalyz.

Capstone Project

One of the most important features of our model is that users (or customers) can retrieve the useful information in real-time manner. Thanks to the power full of Django framework and the specificity of the trained classifier, the relevant information from Twitter is extracted and almost immediately streamed to customer with high accuracy and precision.

In sentiment analysis, we choose humanly annotated datasets to train a classification model for tweet analysis. That ensure the later predictions are based on human perception rather than algorithms.

In conclusion, we presented a simple project aiming to provide pieces of useful information of public mood (here is Twitter users) toward a product or a service (Figure 12). The information is real-time, up-to-date, and highly precise because of specific training of the classifier. Despite its simplicity, the system is robust, scalable and in-production ready. Django framework can switch to work smoothly with all popular data base such MySl or Postgres. In a further improvement, REST API takes charge of interactions with data base. It provides endpoints to work directly with the classifier and is expansible to multiple classifiers or models at once. Via the REST API endpoints, the Administration can easily provide proper privileges to individual staff for content management. In the user interactive aspect, html5 and the latest bootstrap ensure the powerful and adaptive interface, which is perfectly displayed regardless screen sizes and devices. In the future, we will aim to maximize user experience utility with JavaScript and ajax in front-end. We will also improve the analysis in the backend so that the system can provide more useful and statistical information.

Code Repository

The code and files in the project are stored in Github repository at: https://github.com/tdavn/class3406

Demo website can be found at: http://aml3406.pythonanywhere.com/

References

Carpenter, J., Tani, T., Morrison, S., & Keane, J. (2020). Exploring the landscape of educator professional activity on Twitter: An analysis of 16 education-related Twitter hashtags. *Professional Development in Education*, 1-22.

Chollet F. (2018). *Deep learning with Python*. Manning Publications Co.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12), 2009.

Hughes A. and Wojcik S., (2019, August 2). 10 facts about Americans and Twitter. *Pew Research Center*. https://www.pewresearch.org/fact-tank/2019/08/02/10-facts-about-americans-and-twitter/

Kaur, H., & Mangat, V. (2017, February). A survey of sentiment analysis techniques. *In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 921-925). IEEE.

Parveez S. & IriondoSentiment R. (Dec 10, 2020). Analysis (Opinion Mining) with Python — NLP Tutorial. *Towards AI.* https://pub.towardsai.net/sentiment-analysis-opinion-mining-with-python-nlp-tutorial-d1f173ca4e3c#a5cb

Pope, L. (2020, December 16). Comparing VADER and Text Blob to Human Sentiment. *Towards data science.* https://towardsdatascience.com/comparing-vader-and-text-blob-to-human-sentiment-77068cf73982.

Slater M. (2018, March 12). By the numbers: Twitter Canada at Dx3 2018. *Twitter.* https://blog.twitter.com/en_ca/topics/insights/2018/TwitterCanada_at_Dx3.html

Tavoschi, L., Quattrone, F., D'Andrea, E., Ducange, P., Vabanesi, M., Marcelloni, F., & Lopalco, P. L. (2020). Twitter as a sentinel tool to monitor public opinion on vaccination: an opinion mining analysis from September 2016 to August 2017 in Italy. *Human vaccines & immunotherapeutics*, *16*(5), 1062-1069.

Tyagi, V., Kumar, A., & Das, S. (2020, December). Sentiment Analysis on Twitter Data Using Deep Learning approach. *In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (pp. 187-190). IEEE.