Department of Computer Science
University of Hong Kong
Final Year Project

Project Plan

Deep Learning Based Public Sentiment Analysis System on Trending News

Students:

Woo Nathan Hugo
Yeung Sum
1. Project Background

Despite the fact that there are some public sentiment analysis projects using Twitter data, most of the projects are focused on specific topics and issues, such as COVID-19 or the president election (Kausar et al., 2021; Wang et al., 2012). The research and analysis involved in these projects take a considerable amount of time to complete, and we will only be able to obtain the public's opinion on certain topics once the analyses are completed and the paper is published. Apart from the researchers, individuals or some groups of people such as companies or the government, probably want to know the public attitudes and opinions toward certain issues. However, it is very time-consuming to gather and analyze the opinions of a large group of people due to the huge information size, or the use of some traditional methods like questionnaires.

Therefore, in our project, we are going to build an automated system that potentially using many novel technologies, such as Variational Autoencoder, Transformers and pre-trained models for analyzing public sentiments regarding the trending topics in the community. Since this project is mainly built for individuals, instead of having the analysis on certain input targeted topics, our system will automatically pick up some trending news happening in the community for analysis. Moreover, since the process of data acquisition and sentiment analysis for user-inputted topics and our project are almost the same, our project can be further extended for the use of other groups of interest.

2. Project Objective

The purpose of this project is to conduct sentiment analysis on more general popular topics. The most common opinions among a large group of people who are using the social media platform will be extracted, their attitude will be shown by a sentiment score along with the opinion. It allows us to understand the public's thoughts on the daily or weekly popular issues. Furthermore, by utilizing our public sentiment analysis system, researchers can conduct research that requires an understanding of public sentiment, or at least gain some insights, without having to conduct a questionnaire.

To deal with the difficult sentiment analysis tasks which involve Natural Language Processing, various machine learning and deep learning methods will be used. With the help of the latest technologies, we are expecting our models would be performing well in
accomplishing the summary and analysis tasks. Since the whole system will not need human intervention after all the components are built, it will be fast if the computation power is high enough. Users can know the public attitudes and opinions towards trending news easily through our website, which saves a lot of time and human power for doing research or looking through a large number of comments and posts on social media platforms. For the controversial news, since both sides of the argument would probably be shown at the same time, our system can also help individuals in understanding what the people on the opposing side are thinking.

3. Methodology

The system will collect data from multiple APIs and use deep learning and machine learning models to summarize and analyze the data. There will also be a webpage for displaying the result. The content on the webpage can be updated on a daily or weekly basis.

Section 3.1 will first talk about how the data are collected and how they will be used in this project. Section 3.2 will talk about how to evaluate on the performance of the model. Section 3.3 will discuss the proposed method to generate the sentiment analysis by the system. Section 3.4 will discuss the webpage for displaying the result.

3.1. Data Collection and Usage

This project will use the Twitter API, Reddit API and a popular third-party API for Google Trends to collect the data. The data is obtained legally by the APIs and will only be used for this project. The system first uses Google Trends to get the most popular and recently searched keywords. Using the keywords, the system searches and gets the comments that are related to the popular topics on social media platforms, which are Twitter and Reddit.

Instead of using only 1 social media platform to get the comments from the public, both popular platforms Twitter and Reddit will be used. Due to the different content presentation designs in Twitter and Reddit, news will be spread in a different speed and having different lifespan in the two platforms (Priya et al., 2019). Reddit is suitable for getting information at the early phrase of the event and Twitter is suitable for discussing the news and having updates in a longer period since it is easy for the author to extend the Tweets by quoting the previous Tweet. Therefore, the information gain from different
Platforms is different in some way and potentially having some sort of bias towards the issue. In hope of getting a more representative result, multiple sources of comment data will be used, more platforms can be used in future development.

After obtaining the data, the data will be pre-processed and cleaned, such as filtering links or hashtag to increase the effectiveness of training the model (Sahayak et al., 2015). The data will then be split into different ratio of the training, development and testing dataset; and will be evaluated on different models. Their performance will be compared, and the best ratio will be chosen.

3.2 Sentiment Analysis

The system will use deep learning and NLP models to convert the comments to several vectors/representations which represent an opinion towards the topic. When large enough numbers of comments are collected, the system will start to analyze the comments. Since similar opinions will be close in the vector space, the system should know the popular opinions and the system can get and show the popular opinions by directly getting the original sentence or generating sentences from a generative model. Scores will also be given to each keyword for representing if the public is more positive or negative toward the issue. Public sentiments can therefore be derived from the data.

3.2.1 Model Architecture Plan

Since the main task for our deep learning model is to represent an opinion in a n-dimensional vector space, where n would be the hyperparameter which can be adjusted, we hope our model can get the high-level idea from a sentence and represent it in a vector. And for the reason that there should be no available dataset for mapping sentences to abstract ideas or vectors, our project will use an unsupervised way to train our model. Autoencoder architecture would be a good choice since it is self-supervised which does not require labelled data. Oshri & Khandwala (2015) proposed that beside the reconstruction loss which is calculated from the generated output with the input using cross-entropy, one can use additionally encoder loss which compares the hidden representation of the input and generated output, so that the meaning of the input and generated output is similar at least to the model itself. This can probably help a lot for the training process when using a higher ratio of reconstruction loss at the beginning of the training, then increasing the ratio of the encoder loss. There are many variants to the autoencoder in the area of NLP, such as using RNN,
We will be examining different variants and trying to find the best combination. We may also incorporate some pretrained language models like BERT into our model since the language model already can somehow get the meaning of a sentence.

For getting the sentiment, we will be using a method like transfer learning in the field of machine learning. Radford et al. (2017) also used unsupervised way to train a LSTM model for next character prediction using a corpus of 82 million reviews from Amazon. They found a sentiment neuron that is highly correlated to the sentiment of the review. Therefore, since sentiment is probably related to the high-level meaning of a sentence and would affect the output of the decoder, we can also try to find our sentiment neuron in our model using many public datasets for sentence sentiment. After the sentiment of the popular opinions are obtained, we can estimate the total sentiment score toward certain issue with the scores of the individual opinions and number of like or upvote.

3.2.2 Evaluation Plan

In the very beginning of the training, we may need to check the performance of the model manually and adjust it accordingly so that it will be trained in a correct direction. In the later stage of the training, the performance of the trained model will be benchmarked using various evaluation dataset and metrics. For example, BLEU score is used for evaluating the reconstruction task (Oshri & Khandwala, 2015), IMDB review sentiment classification dataset and Microsoft Paraphrase Corpus for evaluating the performance in sentiment analysis (Radford et al., 2017).

3.3 User Interface for Result Delivery

A webpage will be used for delivering the analysis result towards the users for the ease of development and accessibility. React.js and Node.js will be used for building the website. The webpage will show a list of popular keywords, alongside with each keyword, there will be a list of most popular and common opinions from the general public and a sentiment score for showing if the majority of the opinions are positive or negative towards the issue. The data can be updated on a daily or weekly basis after the machine learning models are trained and the pipelines are programmed. There will also be a history page for showing the history data.
4. Project Schedule and Milestones

Figure 4.1 above showcases the schedule of our project. The pipeline for getting the data from Google Trends, Reddit and Twitter has already been set up before the submission of this project plan. From September to February, we will keep collecting data from Twitter for training the language model since we can only get 500,000 tweets per month from the Twitter API.

For experimenting different models like CNN or RNN, it will take roughly 2 months for training different models and making comparison on them. The experiment will start around mid-October because it requires time to collect and clean the data. After deciding which model to use for our project, it will take around a month to train and fine-tune the initial language model.

After that, it will take around 1.5 months to design and build the webpage for displaying the results of our project. At the beginning of February, we will finalize the language model to try to further improve the performance of our language model. At mid-April, the webpage will be deployed by either hosting it ourselves or renting a virtual machine to host the website.
References


