Saint Petersburg State University

Department of Mathematical Game Theory and Statistical Decisions

Su Shimai

Master's Thesis

Neural network methods for messages analysis in user discussions on YouTube

Specialization 01.04.02

Applied Mathematics and Informatics

Master's Program Game Theory and Operations Research

Science Supervisor

Head of Department of Programming Technologies

Dr. Ivan Blekanov

Saint Petersburg

2021

Contents

Abstract

With the burgeoning development of web space, it usually takes a few seconds for the news to be widespread. Especially for some sensitive or widely popular events, they probably draw the attention of a large number of people who create an upward storm of public opinion all around the society. No matter the influence which the storm brings is positive or negative, when it comes to the massive audience, it has to be tracked from all aspects considering the severe consequences it may lead to. At the same time, for journalists, organizations and relevant parties, it's necessary and obliged to flexibly analyze the user's activities and comprehend the general public opinion over specific topics. Thereafter they are capable of making the right response, discovering the actual concerns and even detecting the risk of misleading manipulation.

In this paper, the author mainly proposes a general solution to the analysis of user discussion and user network. And the author chooses YouTube which is one of the most famous and representative social platforms as the data crawling source. Combined analytic modules and different approaches including the state of the art natural language processing models are being applied in our solution. In our experiment, the case of American police misconduct in dealing with a black man is taken as our illustration for how to operate the actual analysis. Temporarily our model only supports English language text.

Introduction

The Relevance of Research Topic

Nowadays, the videos or news with more than millions of comments are quite ubiquitous because of the consistent investment of large scale of Internet infrastructure construction all around the world. According to the Statista, there are roughly 4.66 billion people around the world using the Internet at the start of 2021. This number is close to 60 percent of the total population in the world and it is still climbing. Imagine it, when the latest news are just emerging, they only takes a few minutes or even seconds to be widespread among the colossal Internet user. We are definitely excited to witness such miracle, in another respect, it also demonstrates the great development of modern society. However, we should not let the obvious phenomenon blind our eyes. In some perspectives, the huge public opinion storm attached to these news is a double-edged sword which can severely damage the stability, prosperity and safety of society as well. Especially for the government, the journalists, the company and the relevant parties, they can easily be pushed to the centre of discussion as shown in many real cases. Under this circumstance, they are more eager to figure out the users' feedback [17] through various methods and we are sharing the same thought with lots of people that it's quite urgent to strengthen robust and sufficient ability of sentiment analysis of the public.

Speaking of sentiment analysis, it's also known as opinion mining which has close relationship with natural language processing, text analysis, computational linguistics [21]. When sentiment analysis was first being introduced on public opinion analysis at the beginning of 20th century, it was used on written paper document and 99% of the papers which in-

terpret computer-based sentiment analysis only have been published after 2004 [15]. In particular, recently the techniques related to natural language processing are developing rapidly and the focus of application of sentiment analysis has been turned to Facebook, Twitter and other social platforms. But we found that the actual usage of these researches is not adequate in spite of burgeoning innovations. And what we would like to accomplish in this paper is exactly trying to close the gap and applying the latest models into detecting users' sentiments which is served for better understanding of messages in YouTube.

The Aim and Objectives of Work

The prominent aim of our work is to propose a general solution to the analysis of user message and user network by different events shown in social network. Through such solution, it's possible to detect hidden dependency among users and disclose detailed information. To be frank, currently there are plenty of mature sentiment analysis systems which have already been successfully put in market, such as brand24 and Mediatoolkit. However, in this thesis, we are not trying to bringing up a fresh new model, instead, we pay more attention to the integration of the updated language models based on high performance distributed computing platform in real cases.

Meanwhile, in order to achieve our desired outcome, following steps have been taken. Step A: Investigation of relative techniques and papers. Actually, with the rapid development of computer science within these years, lots of outstanding techniques in language model have been proposed, such as long short-term memory, bidirectional language mode, embeddings from language models, universal sentence encoder, bidirectional encoder representation from Transformers and etc. Step B: These

approaches are being thoroughly compared for making the optimal decision to the usage. Step C: We learn the common complete solution for sentiment analysis from [11]. As described in our solution, we design our architecture starting from data crawling, data preprocessing, combination of modern language model and summarization. In contrast to existing solutions, the most paramount difference is the combination of several modern language model is being put into action. On one hand, these models emphasize different aspects - speed and accuracy which can be chosen according to actual need, on the other hand, through the dual model, we could add an insurance to the result analysis instead of solely based on one model. Besides, compared with the popular analytic principle, we are making an optimization named as separation mechanism to present the analysis result more acceptable and more accurate. Step D: Eventually, we select suitable tools for implementing our solution. Specific descriptions for tools are discussed in next section.

Practical Values of Work

More and more, the analysis of general public opinion has become a prerequisite ability for many areas. For company, they would like to detect the general feedback or principal concern of their new product. For the consultancy, they are capable of previously understanding the trend of striking issues from the analysis. For journalist or social media, the public opinion analysis could do them a great favor when they are trying to make a solid and thorough report on the specific news. As to government, they are able to make precise policy or action to the problem which is disclosed in the sentiment analysis. For instance, the nationwide public opinion detection platform has been widely applied in China. The platform is capable of

detecting the timely primary topic of the public from different kinds of social network. Once the topic and general sentiment are determined, the corresponding person in charge will be informed and they will take right reaction to it. That's why we believe it is beneficial to apply the state of the art of techniques to comprehend the public opinion and provide the evidence for the right reaction. Meanwhile, we also make the search of pivotal figures among users possible with the help of social network analysis.

Our work inherits the basic character of the cases above which indicate that our solution can be utilized for various scenarios as well. Besides, we are more enthusiastic to promote it into other fields where our solution could better serve the people.

Structure of Work

Basically, our solution contains the following 5 steps: data crawling, data preprocessing, analytical module for user message analysis, summarization analysis and social network analysis. In terms of steps, the thesis is divided into 6 parts. Part I, Introduction. We introduce the background, aim and general application of sentiment analysis. Part II, Overview. This part is mainly about the description of current methods for sentiment analysis and the solution which I particularly proposed. Part III, Solution for user discussion analysis in Youtube. It refers to the actual models and solution which are utilized in our project. Part IV, Experiment. We disclose several comprehensive experiments according to our theoretical part. Part V, Conclusion. Part VI, reference.

1 Overview

1.1 Overview of Existing Methods

In order to have a general understanding of the overall framework of unsupervised approach in natural language processing, we made a deep research on the relative methods - Yake [19], TF.IDF [6], TextRank [18], TopicRank [1], optimization of opinion mining for sentiment analysis [14]. As demonstrated in Figure 1.1, from the general view, the rationale of analysis of text can be grouped into statistics, graph-based and embedding. Sentiment analysis is built based on one of them - embedding. In fact, embedding can be separated into word-level and sentence-level, but they are not working in parallel, sentence embedding can only be obtained afterwards.

Beginning from word-level, Word embedding consists of two methodologies, context-free and context-based. The former represents that the generation of embedding of the word is not affected by the peripheral contexts. For instance, the embedding of word *bank* in sentence: I got some cash from bank is totally identical to it in sentence: the bank of river is quite steep. Word2vec is right the representative. As to context-based, apparently, the embedding of word changes under different conditions. When it comes to the sentence embedding, there are manifold techniques existing or recently emerging, such as long short-term memory, embeddings from language model, bidirectional language model, universal sentence encoder, Generative Pre-Training, bidirectional encoder representations from Transformers and etc. Some of them put the accuracy of representation on the first priority, like the method which is constructed on the foundation of RNNs. Some of them prefer to keep a balance between accuracy and time cost. Meanwhile, we are focusing on making an optimal arrangement of

the models.

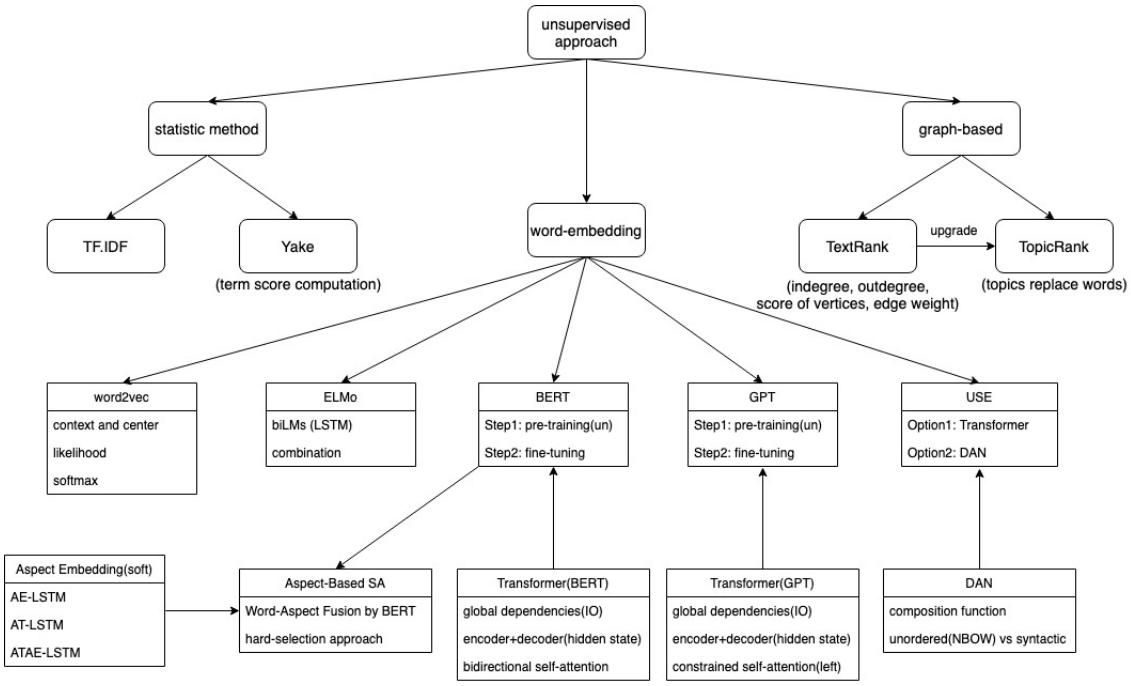

Figure 1.1: NLP Unsupervised Approaches

1.1.1 Embeddings from Language Models

Various to the widely used word embeddings, embeddings from language models (ELMo) [13] word representations are dealing with entire input sentence. They are accomplished with the help of two-layer bidirectional language models (biLMs) with character convolutions, which also indicates that the computation of ELMo is colossal as we'll discuss in section LSTM.

Bidirectional Language Models

Forward: Given a sequence of N tokens $(t_1, t_2, ..., t_N)$, a forward language model computes the probability of the sequence by modeling the probability of token t_k given the history $(t_1, ..., t_{k-1})$:

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, ..., t_{k-1}). \tag{1.1}$$

At each position k, each LSTM layer outputs a context-dependent representation $\overrightarrow{\mathbf{h}}^{LM}_{k,\ j}$ where $j = 1, ..., L$(L: layer of deep model). The top layer LSTM output $\overrightarrow{\mathbf{h}}^{LM}_{k,\ L}$ is used to predict the next token t_{k+1} with a softmax layer.

Backward: a backward language model computes the probability of the sequence by modeling the probability of token t_k given the future context $(t_1, ..., t_{k-1})$:

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, ..., t_N). \tag{1.2}$$

The computation is analogous to the forward language model.

biLM: A biLM combines both forward and backward language model. The objective is to maximize the log likelihood:

$$\begin{aligned} \sum_{k=1}^{N} (&\log p(t_k | t_1, ..., t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \\ &+ \log p(t_k | t_{k+1}, ..., t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) \end{aligned} \tag{1.3}$$

where Θ_x denotes the token representation, Θ_s is Softmax layer.

ELMo

ELMo is a task specific combination of the intermediate layer representations in the biLM. For each token t_k, a L-layer biLM computes a set of $2L + 1$ representations:

$$\begin{aligned} R_k &= \{\mathbf{x}^{LM}_k, \overrightarrow{\mathbf{h}}^{LM}_{k,j}, \overleftarrow{\mathbf{h}}^{LM}_{k,j} | j = 1, ..., L\} \\ &= \{\mathbf{h}^{LM}_{k,j} | j = 0, ..., L\}, \end{aligned}$$

where $\mathbf{h}_{k,0}^{LM}$ is the token layer and $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$ for each biLSTM layer.

For inclusion in a downstream model, ELMo collspses all layers in R into a single vector. In the simplest case, ELMo just selects the top layer, $E(R_k) = \mathbf{h}_{k,L}^{LM}$. More generally, a task specific weighting of all biLM layers is being computed:

$$ELMo_K^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

In this equation, S^{task} are softmax-normalized weights and the scalar parameter γ^{task} allows the task model to scale the entire ELMo vector. As stated that γ is of practical importance to aid the optimization process.

1.1.2 Long Short-Term Memory

The invention of long short-term memory (LSTM) [3] is based on Recurrent Neural Networks (RNNs). Traditionally, a recurrent neural network can be viewed as multiple copies of the same network, each passing a message to a successor.

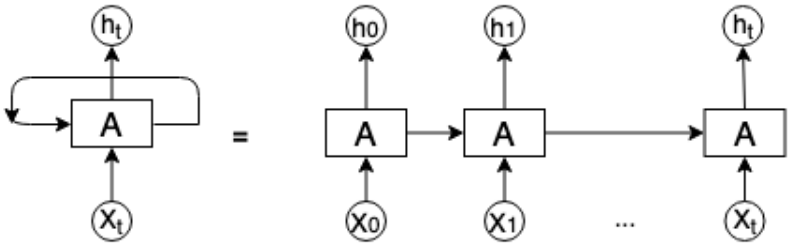

Figure 1.2: Recurrent Neural Network

Theoretically, RNNs should be capable of dealing with long-term dependencies in terms of its recurrent structure. However, in practice, it doesn't show the expected performance. That's why LSTM is specially

designed still as one of RNN to solve this shortcoming. As we can see in Figure 1.3 and 1.4, the prominent difference between standard RNN and LSTM is the construction of their repeating module. Instead of consisting of only one single neural network layer, there are four layers cooperatively working by discipline.

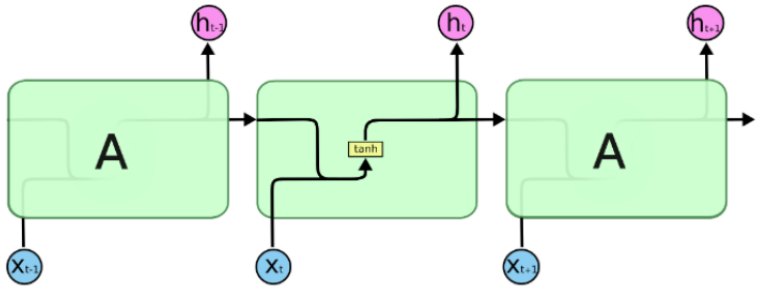

Figure 1.3: The Repeating Module in Standard RNN

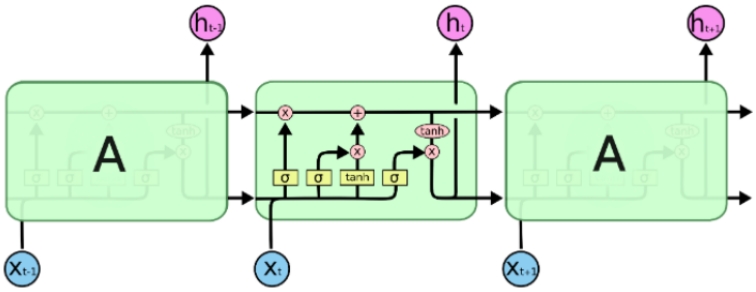

Figure 1.4: The Repeating Module in LSTM

There are many variants of language model from LSTM, such as sentiment classification on attention-based LSTM [22]. Thing is although the long-term dependencies problem is tackled, LSTM still inherits the common character of RNN that the accuracy of information learning is guaranteed, meanwhile, the complexity of computation is enormous greatly shown in its time and memory consuming when it comes to the massive operation. From our prospective, we would firstly go for the model which costs less time and memory. As to other variants including bidirectional language model, embeddings from language models, most of them are realized on

the foundation of LSTM.

1.1.3 YAKE

As far as I know, TF.IDF is one of the most simplest statistical method of feature extraction. For keyword extraction, especially in complex situation, TF.IDF alone is not capable of generating result with high accuracy. On the contrary, YAKE is a light-weight unsupervised automatic keyword extraction which replies on multiple statistical text features which are extracted from text such as term co-occurrence and frequencies. And we will discuss several main functions of it below.

Feature Extraction

Casing: T_{Case}

$$T_{Case} = \frac{\max(TF(U(t)), TF(A(t)))}{\ln(TF(t))} \tag{1.4}$$

where $TF(U(t))$ is the number of occurrences of the candidate term t starting with an uppercase letter, $TF(A(t))$ is the number of times the candidate term t is marked as an acronym and $TF(t)$ is the frequency of t. In this way, the more often the candidate term occurs with a capital letter, the more important it is considered.

Term position: $T_{position}$

$$T_{Position} = \ln(\ln(3 + Median(Sen_t))) \tag{1.5}$$

where Sen_t is the set of positions of the sentences where the candidate term t occurs, and the Median function is the median of Sen_t. Here a double log is used to smooth the difference between terms that occur with a large

median difference.

Term frequency normalization: TF_{Norm}

$$TF_{Norm} = \frac{TF(t)}{MeanTF + 1 \times \sigma} \tag{1.6}$$

where $MeanTF$ denotes the mean of the frequencies. σ is the standard deviation. By applying normalization, it can prevent a bias towards high frequency.

Term relatedness to context: T_{Rel}

The aim is to determine the dispersion (D) of a candidate term t with regards to its specific context. According to Machado et al. (2009) who state that the higher the number of different terms that co-occur with the candidate term t on both side, the less significant term t will be. And $DL[DR]$ is formalized in:

$$DL[DR] = \frac{|A_{t,w}|}{\sum_{k \in A_{t,w}} CoOccur_{t,k}}$$

where $|A_{t,w}|$ represents the number of different terms which occurs on the left or right side of a candidate term t within a window size w, in respect of the number of k terms that it co-occurs with. The final equation of T_{Rel} is given by

$$T_{Rel} = 1 + (DL + DR...) \times \frac{TF(t)}{MaxTF}. \tag{1.7}$$

Term different sentence: $T_{Sentence}$

$$T_{Sentence} = \frac{SF(t)}{\#Sentence} \tag{1.8}$$

where $SF(t)$ is the sentence frequency of the candidate term t and $\#Sentence$ is the total number of sentences in the text.

Computing Term Score

The previous 5 feature weights are being combined into a $S(t)$ score. The smaller the value, the more significant the 1-gram term is,

$$S(t) = \frac{T_{Rel} \times T_{Position}}{T_{Case} + \frac{TF_{Norm}}{T_{Rel}} + \frac{T_{Sentence}}{T_{Rel}}} \tag{1.9}$$

Basically, the keywords are obtained through such combination of statistical features afterwards.

1.2 Application and Solution

As we've found that the sentiment analysis has been packaged as a quite mature and popular product in the market.

Platform	Capability	Price	Limitation
Brand24	instant access	99$	low-weight
Mediatoolkit	comprehensive applications	7242$	expensive
monkeylearn	classifier and extractor	299$	small scale
Talkwalker	real-time monitoring	750$	insufficient analysis
Repustate	multi-linguistics	299$	narrow usage
Lexalytics	business usage	2500$	limited aspects

Table 1.1: Comparison of Existing Solution

As shown in the table above, we found that current sentiment analysis platforms are not equipped with a complete system for full-aspect analysis. And because of obvious obstacle of black box, we can't penetrate the specific models which they utilize in their system. However, from my perspective, considering the high requirement of speed and memory, normally they won't use extremely complicated models. And they won't make multiple models run in parallel neither. That's why I believe there is man-

made upper bound which would somehow restrict the result. The full-cycle solution which we propose is specially designed for the local environment which need to deploy the distributed computing platform, database system and relative modules. Based on such environment, two combined models are being set to satisfy the need of users. In addition to the models, we also largely approach to better presentation of result. Separation mechanism is being applied to thoroughly detect the evolution of the public sentiment and the user could have a broad view of discussion network with social network analysis which we make as a supplement.

2 Solution for User Discussion Analysis in YouTube

There is one thing which need to be clarified that it is not possible to bring the text into the model at once. Following the common sense, from word vector to sentence embedding, this is the obligatory process. Before implementing the models, we took a deep research on the NLP methods in term of unsupervised approaches. The disadvantages and advantages are all being considered to find the optimal solution for our project. As described in Figure 1.1 in which word2vec, USE and BERT are finally chosen as our foundation.

2.1 Architecture of Solution

The general frame of my solution is organized as follows, data crawling - data preprocessing - analytical module for user message analysis - summarization.

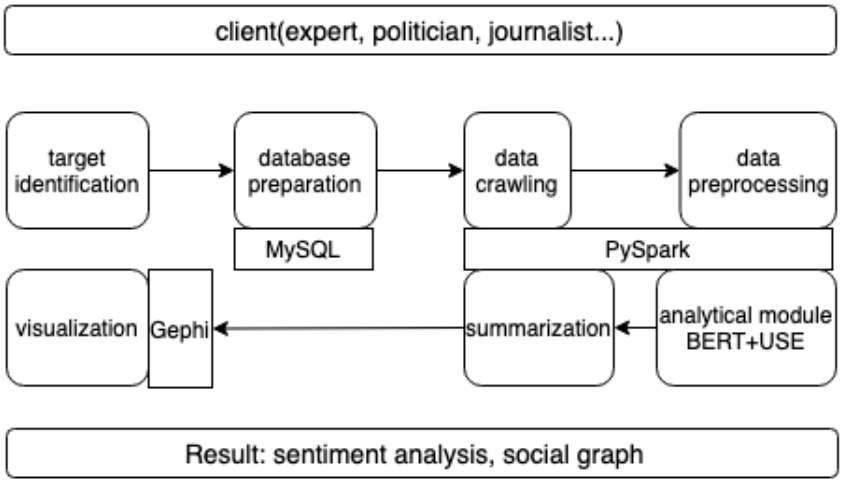

Figure 2.1: Architecture of Solution

According to our plan, we initially confirm the relevant data which we have interest in and design following tables and parameters. New parameters can be added into the table later.

Parameter	Type	Length	Primary key	Null
id	int	auto	True	False
sign	int	auto	False	False
userName	varchar	255	False	False
channelId	varchar	255	False	False
comment	varchar	6000	False	True
publishtime	varchar	255	False	False
likeCount	int	auto	False	True
replyCount	int	auto	False	True

Table 2.1: CommentsVideo

Parameter	Type	Length	Primary key	Null
id	int	auto	True	False
PN	varchar	255	False	False
comment	varchar	255	False	False

Table 2.2: SentimentAnalysisDataset(test+train)

2.1.1 Data Crawling

Two different methods of crawling are being applied in our research - Selenium and YouTube Data API.

Selenium

In tradition, Selenium is a specific tool which is used for automation test of Web. But crawling data from YouTube shares similar procedures. After triggering the code, CSS selector works as the element hunter locating the target which we highlight in HTML source. And once it finds the target, we make the data written into the database directly. The obvious drawback of this method is that the regular operation can probably be interrupted due to low caching speed, unexpected windows or failure searching. And

such method is not the optimal one to deal with large amount of data with respect to its long time cost. Therefore, we don't suggest such method unless under some special conditions.

YouTube Data API

Compared with Selenium, YouTube Data API, with strong robustness and low cost time, it can flexibly access the Google server to request the response which conveys the data. As to the actual usage, first of all, we obtained the developer key after applying for authority in Google developer. With developer key, we are authorized to build the connection with Google server and make the communication. Back to the parameters we've determined previously, we inform Google server the request which contains the corresponding parameters. Google server successfully receives the request and send back the data.

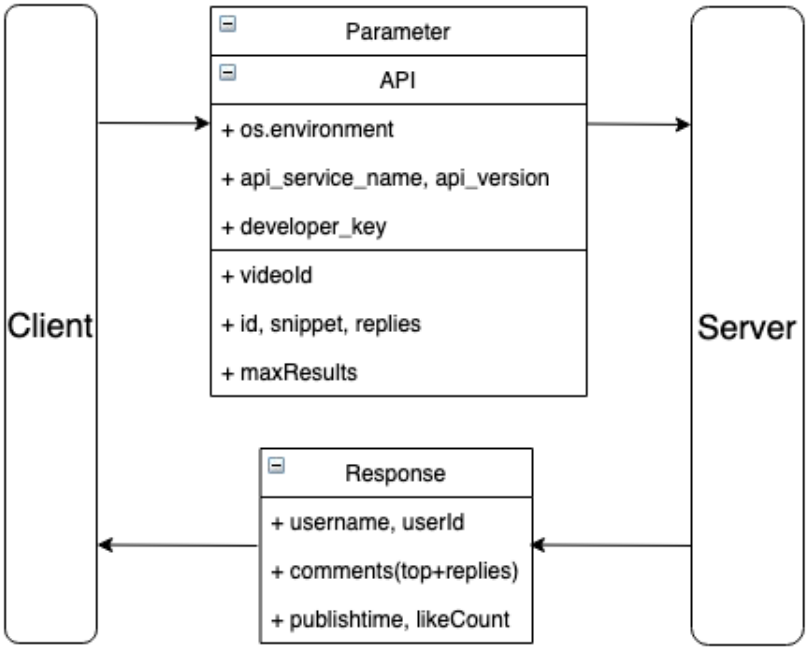

Figure 2.2: YouTube Data API

2.1.2 Data Preprocessing

The datailed procedures of data preprocessing are manifold corresponding to the purpose of task. The performance of these procedures turns to affect the final result. For information retrieval, the data preprocessing contains tokenziation, lemmatization and etc. On the contrary, for sentiment analysis, it concentrates on the understandable integrity of the meaning. Commonly, the data which is crawled from YouTube is not in perfect case. Letters are not the only format which can be expressed online. For the sake of high accuracy of training and large effort of removal of noise, it is more suitable to preprocess the data before being input into the model. The Regular Expression(regex) is taken to fulfill this task given its feasible combination of sequence of characters.

Irrelevant Symbols

As we can imagine and also what we've found in the real case, emojis, links, non-English letters in our case and other irrelevant symbols appear in the comments. We solved this problem by constraining the sequence of search pattern within $[A-Za-z]$ and filtering the fixed pattern for link recognition in regex.

Stopwords

Stopwords, usually, are the most common words in a language which will be filtered out in preprocessing. The reason of application of stopwords filtering is to pin down the important parts of text. However, in our project, we are possessed of different aims which we have to highlight and maintain the original meaning of comments. If we apply stopwords into my experiment, the meaning of comments can be dramatically altered, for instance,

sentence 'I don't suggest to make such action' probably will be truncated into 'suggest action'. So in this way, such technique has to be abandoned.

Group Series

Out of the need of analysis of evolution of public opinion, we designed this group series. In contrast to the standard technique which classifies the texts by day or week, we proposes the idea of group series in which the texts are grouped by a changeable group size. The paramount advantage of such method is that all analytic units are formed in same dimension. Considering we form the analysis by each day or week, the figure would just indicate the situation on each unit regardless of the total amount. And if we try to normalize the count, it will neglect the number of participants. Besides, we noticed that the distribution of comments for each day is extremely uneven. Through the application of group series with movable window size, the curve demonstrated in the figure remains in equal level.

2.2 Architecture of Technique Stack

The version of different platforms or applications has to be matched because of the possible existence of incompatibility.

Java version: Java8

Distributed computing platform: Pyspark-3.1.1

First of all, considering the huge amount of training dataset, distributed computing platform is our first choice and we choose Spark as our platform for computation due to its appropriate design and good performance. The prominent character of Spark is its resilient distributed dataset which can symbolize the dataset and avoid unnecessary computation during the running process.

Environment:Python-3.7, SparkNLP-3.0.2

The language models are implemented with help of SparkNLP which is a outstanding natural language processing library built on distributed computing platform Spark. It contains lots of state-of-the-art models which can be easily merged into the project. This is also another reason why we choose Pyspark as our computing platform.

Editor: Pycharm

Pycharm is a comprehensive and powerful editor for python coding.

Database: MySQL

The standard tool for database store and management is MySQL.

Social network analysis tool: Gephi

We prefer to utilize Gephi to make the visualization of users network. There are many layout algorithms in Gephi, we are able to generate different graphs corresponding to actual requirements.

2.3 Neural network approach for user message analysis

We talk about how the approaches which we use in our solution theoretically work and the subsections below are sorted in a logical way.

2.3.1 Word2vec

Actually, there are two types of word modeling, context-free and context-based. The former one means that the calculation of word vector is not affected in term of the different peripheral words. And the meaning is right opposite to the latter. As to our research, we opt to use the context-free modeling (word2vec) [4] with respect to its less computation and low time cost.

Word2vec was initially designed and implemented by Tomas Mikolov

and his google team. The rationale of it is that a word's meaning is given by the words that frequently appear close-by. Based on such idea, given a sentence, the center word is represented as W_t, the peripheral words are denoted by W_{t+j} or W_{t-j}, where j is the distance between center word and current one. Symbol $+, -$ depends the current word is on the right or left as shown in Figure 2.3.

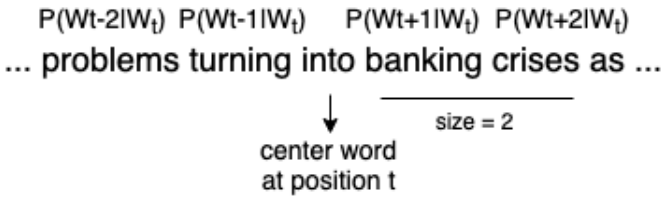

Figure 2.3: word2vec rationale

Since the center word is decided by the surrounding words, the conditional probability is inevitable to take into account. Based on the such modeling, likelihood function is being constructed and the aim is to maximize it to get close to the samples,

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} P(W_{t+j}|W_t; \theta),$$

and for simplicity, the maximization of likelihood function can be transformed into another form, it turns out to minimize the objective function:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m, j \neq 0} \log P(W_{t+j}|W_t; \theta). \tag{2.1}$$

The calculation of $P(W_{t+j}|W_t; \theta)$ is solved in Softmax Function. Softmax function is frequently used in deep learning. It mainly maps arbitrary values to a probability distribution. The v_w denotes the vector for center

word c, u_w is vector for context word o. Therefore,

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)} \tag{2.2}$$

in which V represents the word in text corpus. To minimize the objective function, the calculation of partial derivative on v_c is formed,

$$\begin{aligned}
\frac{\partial}{\partial v_c} \log \frac{exp(u_o^T v_c)}{\sum_{w=1}^{v} exp(u_w^T v_c)} &= u_o - \frac{\partial}{\partial v_c} \log \sum_{w=1}^{v} exp(u_o^T v_c) \\
&= u_o - \frac{1}{\sum_{w=1}^{v} exp(u_w^T v_c)} \frac{\partial}{\partial v_c} \sum_{x=1}^{v} exp(u_x^T v_c) \\
&= u_o - \frac{1}{\sum_{w=1}^{v} exp(u_w^T v_c)} \sum_{x=1}^{v} exp(u_x^T v_c) \frac{\partial}{\partial v_c} (u_x^T v_c) \\
&= u_o - \sum_{x=1}^{v} \frac{exp(u_x^T v_c)}{\sum_{w=1}^{v} exp(u_w^T v_c)} u_x = u_o - \sum_{x=1}^{v} p(x|c) u_x.
\end{aligned}$$

And the rest can be completed by gradient descent.

2.3.2 Transfer Learning

Before referring to the detailed language model for sentence, Transfer learning [12] has to be comprehended in advance. Transfer learning is usually expressed in the form of pre-trained models which have already been created to solve a similar problem. The straight benefit of using pre-trained model is that we don't need to train the model from scratch again, which would save us enormous time and efforts to reinvent the wheel. Similar to the process of knowledge absorbing in which we learn things from someone who is more professional than us, based on such knowledge, we will be operating in other fields which are not equivalent to the original downright. For instance, we could invoke the well-equipped package instead of spending huge time training the model to recognize animals. Also the additional

function like the recognition of new species can be supplemented based on the current model. As indicated in Figure 2.4, the knowledge which is obtained from task 1 can work on a close task 2 smoothly.

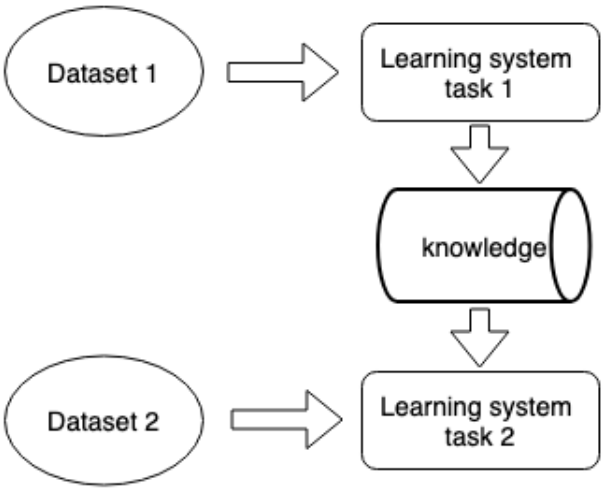

Figure 2.4: Transfer Learning

From the perspective of mathematics, for a domain $\boldsymbol{X} = \{x_1, ..., x_n\}$, the task is defined as $\boldsymbol{Q} = \{Y, F\}$. $\boldsymbol{Y}$: label space (according to actual situation), $\boldsymbol{F}$: predictive function which is learned from the feature vector and label pairs $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$.

If there are two domains represented by source domain D_S and target domain D_T, given a source domain D_S with a corresponding source task Q_S and a target domain D_T with a corresponding task Q_T, transfer learning is the process of improving the target predictive function F_T by using the related information from D_S and Q_S, where $D_S \neq D_T$ or $Q_S \neq Q_T$.

2.3.3 Universal Sentence Encoder

The description of Universal Sentence Encoder(USE) [8] is just exactly what its name shows. This method is utilized for transforming the sentence into fixed dimensional embedding. According to the official docu-

ment, there are two types of approaches to achieve final sentence embedding. One is Transformer which we would discuss in the next section. In our research, we focus on the another approach, deep averaging network(DAN) [16]. The reason why we opt for the second is that although Transformer demonstrates high accuracy, it also takes greater complexity and resource consumption as the price. Meanwhile, DAN makes a good balance between the accuracy and efficiency. Hence with the consideration of optimal solution, DAN is settled. As to why we still use Transformer in the next section? In the control experiment, we have to make sure that one of them could probably give us a trustable guarantee.

Deep Averaging Network

The vector representation z for input text $\mathbf{X}$ is computed by averaging the word vector $v_w \in \mathbf{X}$. For each layer, presented in Figure 2.5,

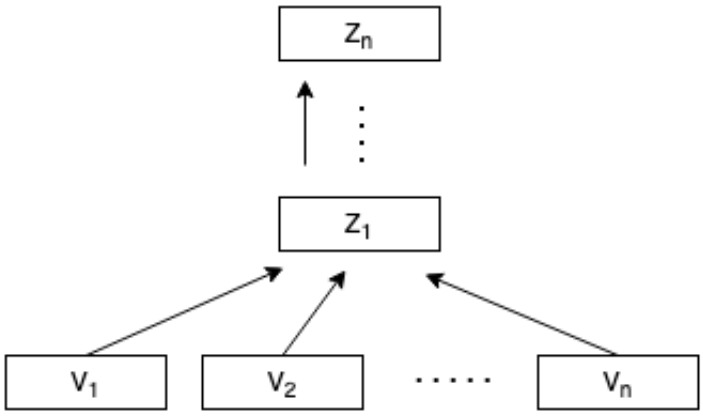

Figure 2.5: DAN

And z_i can be organized in the following way, g represents averaging function,

$$z_i = g(z_{i-1}) = f(W_i z_{i-1} + b_i), i \in [1, \ldots, n]. \tag{2.3}$$

Then the softmax layer takes the final layer's representation z_n as input for

prediction. And the model is pre-trained by minimizing the cross-entropy error, which for a single training instance with ground-truth label y is

$$L(\hat{y}) = \sum_{p=1}^{k} y_p \log(\hat{y}_p)$$

where k is the number of types of labels, $\hat{y} = \text{softmax}(z_n)$.

2.3.4 Bidirectional Encoder Representations from Transformer

Bidirectional encoder representations from Transformer [10] is abbreviated to BERT, which is published in 2018. And once it comes out to the world, the significant performance makes it swiftly dominate the encoder field in NLP. Back to Figure 1.1, there is one quite similar technique called Generative Pre-trained Transformer(GPT) [2]. However, GPT is not equipped with bidirectional self-attention, and this distinguishable difference makes these two techniques differ in their performances. Apart from GPT, we also compare BERT with model of LSTM, which is a popular method in learning the language model. And the model shows that LSTM has more complicated computation and large memory cost due to its recurrent neural network design.

Transformer with Attention Mechanism

Transformer is the backbone of BERT. The highlight of Transformer is that the number of operations required to relate signals from two arbitrary input or output positions is reduced to a constant number which greatly shortens the cost. The effect of solution is being affected, but the multi-head attention whose attention mechanism is equally the core of Transformer compared with relationship between BERT and Transformer, compensates

to the effective solution. In Figure 2.6, we can notice that Transformer consists of two parts, encoder on the left and decoder on the right. With different numbers of layers of Transformer blocks L and hidden size H, BERT is classified into various types as shown in Table 2.3. In our case, $L = 12, H = 512$.

	H=128	H=256	H=512	H=768
L=2	BERT-Tiny	2/256	2/512	2/768
L=4	4/128	BERT-Mini	BERT-Small	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	BERT-Medium	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	BERT-Base

Table 2.3: BERT Model

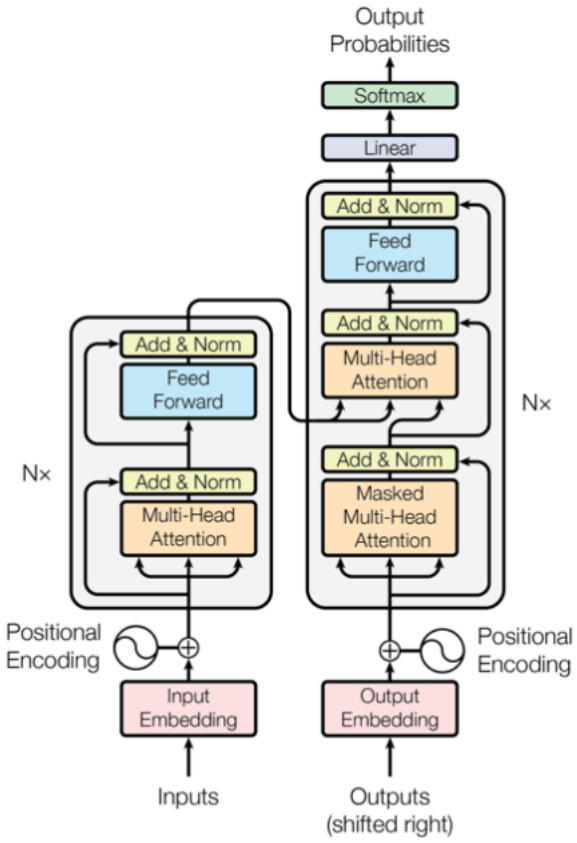

Figure 2.6: Transformer

The purpose of attention mechanism [5] is to draw global dependencies between input and output without the entanglement of recurrence and convolutions completely. Starting from scaled dot-product attention, an attention function can be described as mapping a query and a set of key-value pairs to an output. The Output $\mathbf{C}$ by definition is a weighted sum of the value:

$$\mathbf{C} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2.4}$$

where Q - a query(vector representations of one word in the sequence), $K - V$ represents key-value(vector representations of all the words in the sequence), d_k, the dimension of queries and keys, in our case, $k = 64$.

Learning from scaled dot-product attention, the researchers found that if the attention mechanism is built in parallel with multi-head attentions, it shows better performance than single one. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O \tag{2.5}$$

where $head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, the projections are parameter matrices$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_K}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}} (d_{model}/h = 64, h = 8)$.

2.4 Deep Learning in Sentiment Analysis

In other words, the final step of sentiment analysis is ultimately a class of classification problem. Our goal is to classify the comments into three types, positive, neutral and negative. As we know that there are manifold classification algorithms being promoted currently especially with neural

network framework and we are in favor of deep learning [9], because lots of experiments demonstrate that deep learning has excellent performance compared with other techniques. The primary character of deep learning is that it contains more than one hidden layer in contrast to usual neural network. Moreover, the deep learning which we apply in our case is specifically configured to comply with sentiment analysis, i.e. it could only generate three types of result for any input. Figure 2.7 shows a instance of neural network and the operation of deep learning is identical to it.

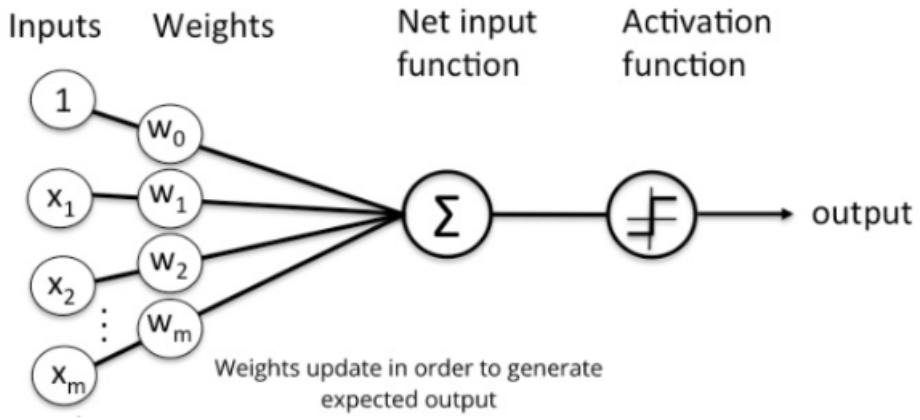

Figure 2.7: Instance of neural network

Input of hidden layer $L2$ consists of the sum product of the weight and input value of input layer,

$$\theta_j^{(2)} = \sum_{i=1}^{p} [x_i * w_{ij}^{(1)}] \tag{2.6}$$

Output of hidden layer $L2$ which utilizes sigmoid function:

$$\alpha_j^{(2)} = \frac{1}{1 + e^{-\theta}}. \tag{2.7}$$

And the adjustment of weight is operated through the back propagation. The actual calculation is presented as follows.

1. Error calculation,

$$E_i = \frac{1}{2}(target_i - y_i)^2,$$

$$E_{total} = \sum_{i=0}^{k} E_i,$$

where k is the number of output, in our case, $k = 3$.

2. Weight updates through chain rule, for instance, $w_{12}^{+} = w_{12} - \eta \frac{\partial E_{total}}{\partial w_{12}}$.

2.4.1 Text-To-Text Transfer Transform

The intention of applying Text-To-Text Transfer Transform(T5) [7] is primarily for summarizing the keywords of the comments. Apparently, YAKE shares same purpose with T5. The reason why we use latter one is that YAKE is much better to deal with big trunk of text instead of short text. And T5 is also constructed based on Transformer which is identical to the one used in BERT. Just a little converse to Bert, T5 is built based on standard encoder-decoder Transformer and it's open for mainly 5 tasks which include translation, question answering, classification, regression and summarization. The way to trigger the model's operation is that adding a task-specific (text) prefix before the original text. In our case, we concentrate on the summarization task in which a prefix-summarize is added to the beginning of text.

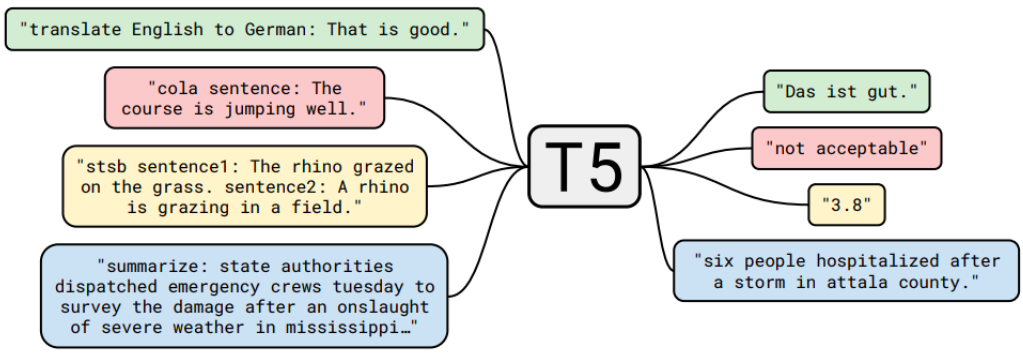

Figure 2.8: Task of T5

3 Experiment

3.1 Description of Dataset

Our dataset is consisted of two types - labeled dataset and real dataset. The labeled dataset which is used for model training is downloaded from Twitter sentiment analysis training corpus dataset (Year 2012) in which we take 230610 comments and corresponding labels. As to real dataset, we decide to get it from YouTube which represents as one of the most representative social platform, attracts massive access and view everyday. With its popularity and convenience of data access, we are capable of crawling users' comments which could demonstrate various ideas. And the video to the real dataset is about the case of misconduct of American police in dealing with black man (YouTube video id=XsgryiPK2is). The earliest user's comment can be tracked to 25th Aug, 2020 and the latest one happened on 15th Oct, 2020. From this period, 4583 comments are crawled and separately stored in the database as we discussed previously.

3.2 Training

The training primarily contains two parts. First part is the evaluation of the performance of two models on the training dataset. Through it, we could make a decision on which model is more reliable. And then we let the model run on the real case to obtain the result.

3.2.1 Model Evaluation

We don't have the settled and right label for real dataset because of the speciality of unsupervised approaches. Under this circumstance, in order to quantify the actual performance of models, the evaluation of model can

only depend on the training dataset. Following the standard principle, we split 80% of training dataset as the training dataset, 20% as the testing dataset to evaluate the performance of different models.

Model	Epoch	Acc	Rate	Speed
USE	20	0.7547	0.0005	Around 0.7h
BERT	20	0.7676	0.0005	Around 3.5h

Table 3.1: Model Comparison)

As indicated Table 3.1, after adjusting the epoch and rate, we approached our model to the best accuracy as good as possible and circumvented over-fitting. It shows that the accuracy of BERT is slightly higher than USE. Therefore, we will more reply on BERT to make the final judgement. However, we can also notice that two models are fairly high and close, in this case, we conclude that both model will be applied in the real case. Beside, for the time cost, USE significantly spent much more less time than BERT. We believe that in production environment, people probably would opt for fast handling due to its close accuracy.

3.2.2 Application of Trained Model on Real Case

The result of application of trained model on real case is presented in the form of evolution of public opinion with ascending time. As described in separation mechanism, we basically divided our comments into 46 groups by time. And there is a little trick here, for the number of last group, it doesn't meet 100, but we still let them be a group, just in the analysis, the actual curve of last group should not be treated equally to others.

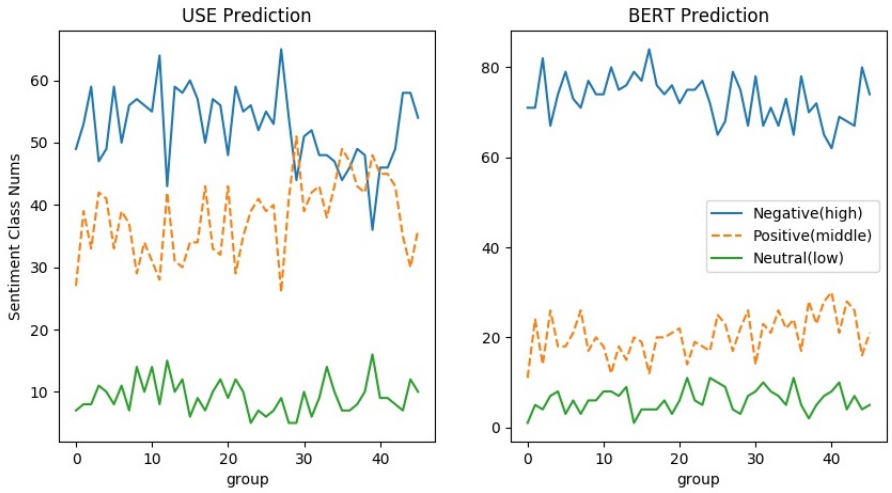

Figure 3.1: Result: USE and BERT

As we can see in Figure 3.1, the general view is similar that the negative sentiment takes the majority, just opinions of few people are neutral. But the curves which are generated by these two models are quite different in detail, in particular, the plot of positive and negative part. The result of USE shows that there is fierce opposition over the topic. However, in the graph of BERT, the negative group is dominating the whole discussion. Since the accuracy is quite close, we can't be 100% sure of the result of BERT. When we are embedding in such scenario, how we can solve it? This solution can be even summarized as a general method to more cases. Hence we will leave this part to the future research.

Besides, we pin down multiple important topics of their discussion as drew in Figure 3.2 as well. The large the word is, the more important it is. Police, cop, shot, john avlon(journalist), lz granderson(journalist), shown as the first priority, so we can basically make sure that currently people are more concerned about the aspect of police instead of the ethnic problem, the right of black man, even the control of gun. And gun, racist, people, time, knife, law, car are just followed. Actually I am not satisfied

with current result, there are still too many trivial words in our plot and I believe there is more optimizations can be done in the future.

Figure 3.2: Summarization of Context

3.3 Visualization of Social Graph by Real Case

Actually, sentimental analysis is just a subset of a complex analytic system. Although the analysis of public opinion is the most prominent task, users who are behind the comment ultimately play a significant role in the network.

The analysis to the users in social network is very sophisticated and it's quite unpredictable to figure out the genuine role of the user in the network if we just take a peek to the comment the user made. He maybe just an ordinary person who leaves random comments without any special intentions or he, who is possessed of unexposed incentive, tries to lead the public opinion and possibly alter the current organization or other cases. Therefore, we suppose that the analysis of user discussion has to be extended to the area of social interaction between users. Social network analysis (SNA) can be interpreted as the process of investigating social structures through the use of networks and graph theory. Commonly, social network forms in the combination of nodes and edges as shown in Figure 3.3.

The general formula for graph is $G = (U, E)$. In Figure 3.3, the node U is represented by the user, the edge E between different notes shows that there is interactive discussion and the broader the line is, the more intensive interaction between them. To acquire the list of node and edge, several works have been used. First of all, there are lots of repetitive username in our database due to its multiple replies and we only leave one in the list. As to the computation of edge, two steps are needed. Initially, the identification of users who involved in the interactive discussion is the prerequisite of the calculation of the strength of link between them. In fact, when we were crawling data, we deliberately designed a parameter *sign* which indicates the level of current comment, 1 : top level comment, 0: second level comment (which is direct reply to top level comment). By following this principle, we are able to identify the involved parties in this conversation. As the result, generally, the network is consisted of 2583 nodes and 1284 edges.

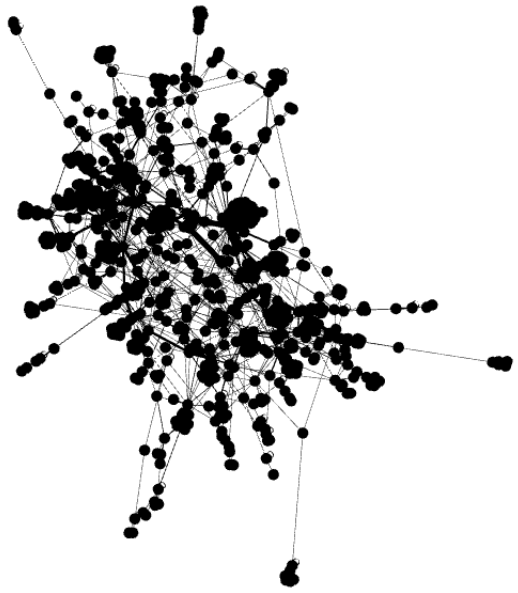

Figure 3.3: Graph of User Discussion Using ForceAtlas2

In addition, based on the previous result of users' sentiment in BERT,

we make a classification of different groups which show various level of sentiment. The weight of neutral sentiment is set as 0.5, positive as 1, negative as 0. Thereafter, we formulate a way of user's sentiment calculation.

$$U_{sentiment} = \frac{1}{N} \sum_{i}^{N} mess_{U,i},$$

where U - current user, i - ith message of user and $mess$ - the sentimental weight of message. And then we convert the message analysis to graph analysis as follows,

$$f(x) = \begin{cases} green, & [0.75, 1] \\ yellow, & [0.5, 0.75] \\ orange, & [0.25, 0.5] \\ red, & [0, 0.25] \end{cases}$$

And based on such principle, we classify different levels of sentiment into 4 groups, green - more positive, yellow - positive, orange - neutral, red - negative. After applying ForceAtlas2 layout algorithm, we obtain Figure 3.4. From Figure 3.4, it's quite obvious to find out that the negative sentiment to the black man was killed by American police dominates the general graph. But generally positive nodes (yellow, green) take a relatively high proportion in the network, we have reason to doubt that there is a symbol of initial stage of polarity which means that the ideological conflict among users is quite salient and this is indeed a dangerous signal which has to be tackled seriously.

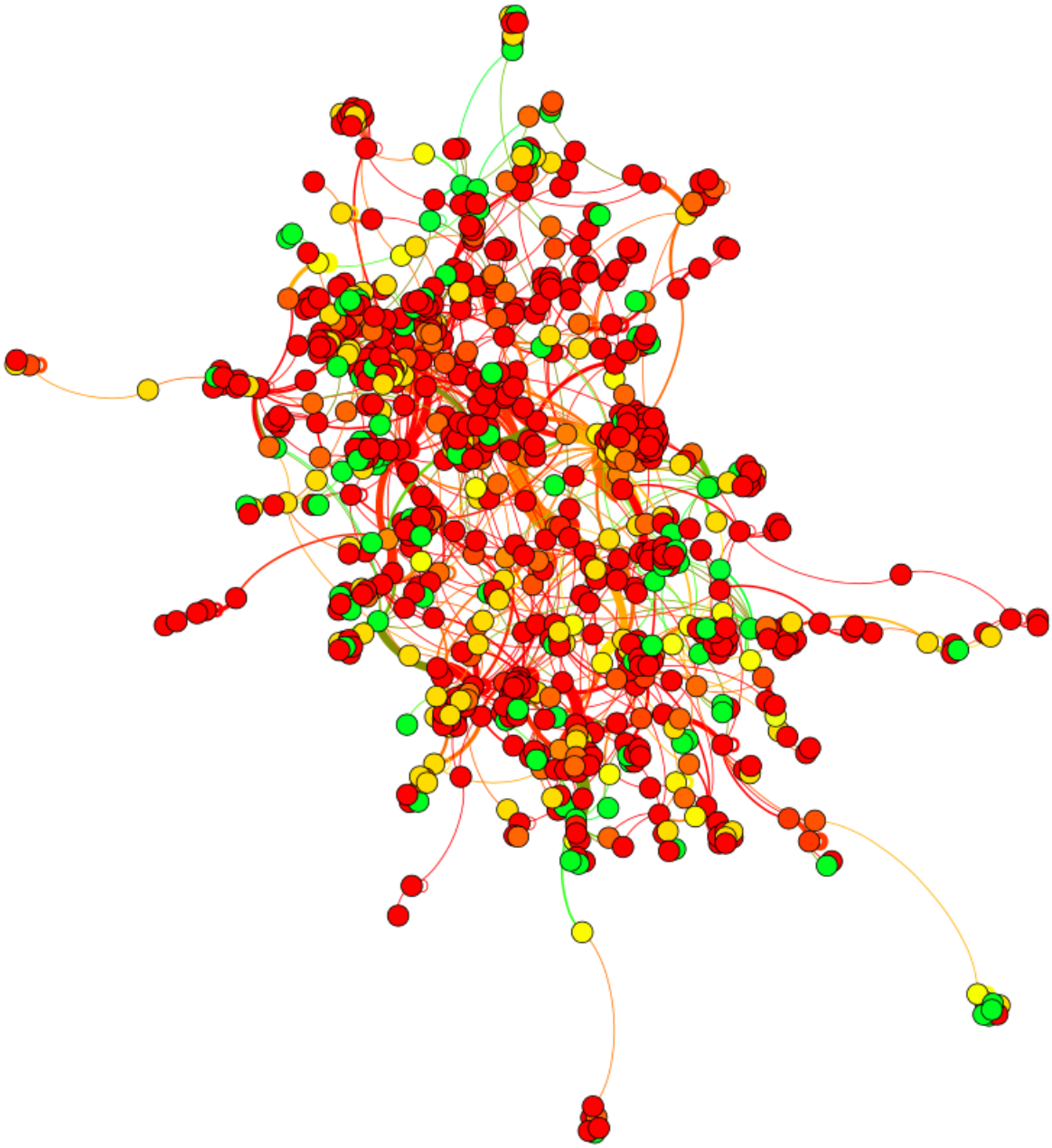

Figure 3.4: Graph of User Discussion with Sentiment labelling Using ForceAtlas2

Meanwhile, It's quite distinguishable to notice that one of the most obvious characters of the generated graph is that there are enormous independent nodes which means that the user has no communication with other users in the peripheral area. Figure 3.5 only presents a small segment of the independent nodes. However, we could easily capture that the negative sentiment occupies most of surrounding nodes.

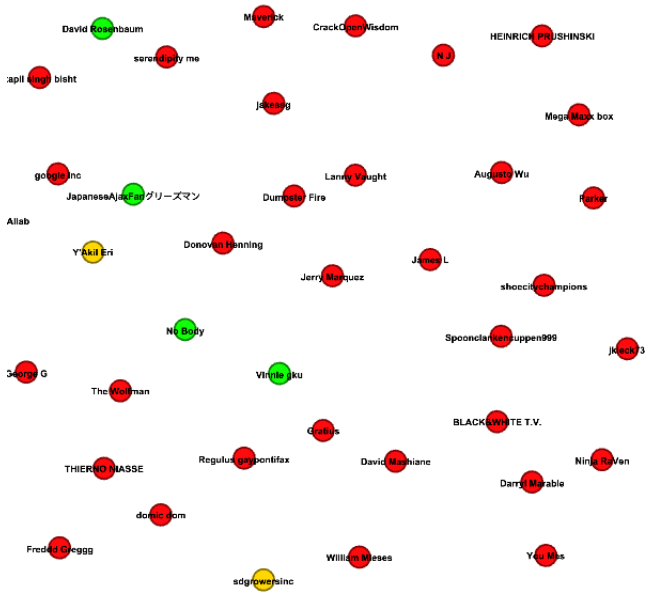

Figure 3.5: Peripheral Users

We also can locate the most crucial segment of the network. As demonstrated in Figure 3.6, the central user draws too much attention from others. An argument regarding to the event must have taken place and from the figure, it's convincing that the negative sentiment which is against the police operation had other beat.

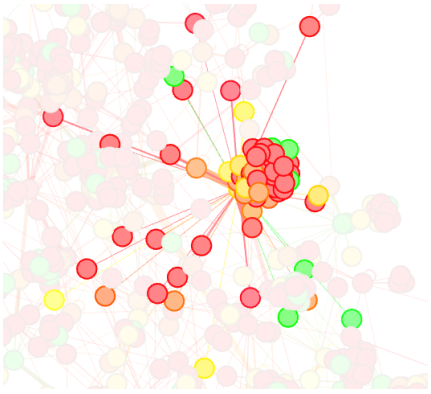

Figure 3.6: One Important Clot in Network

Conclusion

Result of Work

Basically speaking, the primary purpose of our project has been successfully accomplished. We adequately build a modern solution for full-scale scenarios to sentiment analysis in YouTube. Through our solution, people could have a complete knowledge of their interested events, no matter the whole sentiment, the summarization of keywords, or the analysis of social graph. Most importantly, the latest model and double insurance are being involved in this solution which makes the result more trustable.

Future Research

During this period, we find that it is quite difficult to make accuracy and speed both reach their best stage. Commonly, when we maximize one of them, it's inevitable to neglect another. Although the high-performance computing platform is adequate, it doesn't need much time to complete the training process, it's still a great challenge when it comes to massive operation. Meanwhile, we do make a compromise in the balance between accuracy and speed as well, as described previously, USE largely keeps its accuracy when it tries to reduce time cost. However, there is remaining question after obtaining the close performance of these two models, the result which generated on the real case is not close. Under this circumstance, I suppose we can't just simply bring another model to verify it and I am considering to utilize the sampling method which would choose part of comments randomly to check the real sentiment manually. By such sampling, it's possible to provide a confidence level to the model separately which will impact on our decision for sure.

Acknowledgements

Writing these words literately means my graduate study is about to finish. Honestly speaking, coming this long way, I was so into the works and dealing with various situations, meanwhile neglecting surrounding magnificent view. This time I've got a precious opportunity to give myself these three years with one year preparatory studying a conclusion. The scene of first time when I got off the plane and stepped on the land of Russia is still vivid in my mind. I remember I was totally like a immature calf rushing to make myself familiar with completely unknown environment.

Along this transformative path, I am so grateful that I've met lots of wonderful friends, Berk, Mateen, Ye Litian. They helped me a lot no matter in lives or study when I was having trouble of making my things sorted initially. I couldn't thank more to them sincerely. And I'm not trying to exaggerate the difficulty, but I did live under well protection of my parents before. I couldn't imagine that I would tackle all the problems on my own one day, while I'm also quite delighted that I've made excellent preparation for graduate study eventually.

I truly appreciate that I could have the chance to complete my graduate study under the supervision of Dr. Ivan Blekanov. He consistently gave me many valuable ideas to my paper and I am greatly sorry that I couldn't implement all of them. However I deeply hope that I am able to proceed future research with the professional attitude you've shown me. Last but not least, I own them, people who impressed me with their professional skills, who taught me with their profound knowledge, who helped me a huge thanks. Thank you.

References

1. Adrien Bougouin, Florian Boudin and Beatrice Daille. 2013. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543-551.

2. Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. In *arxiv*.

3. Alex Sherstinsky. 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, Volume 404.

4. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. 142-150.

5. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Adian N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 6000-6010.

6. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze. 2009. Introduction to Information Retrieval. Cambridge University Press.

7. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu. 2020.

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.*Journal of Machine Learning Research 21 (2020)*, 1-67.

8. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope and Ray Kurzweil. 2018. Universal Sentence Encoder. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169-174.

9. Ian Goodfellow, Yoshua Bengio and Aaron Courville. 2016. Deep Learning. MIT Press.

10. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171-4186.

11. Jalaj Thanaki. (2017). Python Natural Language Processing. Packt Publishing.

12. Karl Weiss, Taghi M. Khoshgoftaar and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data*, DOI 10.1186/s40537-016-0043-6

13. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clarkm Kenton Lee and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *NAACL*.

14. Mengting Hu, Shaiwan Zhao, Honglei Guo, Renhong Cheng and Zhong Su. 2019. Learning to Detect Opinion Snippet for Aspect-Based Sentiment Analysis. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 970-979.

15. Mika Mantyla, Daniel Graziotin and Miikka Kuutila. 2018. The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers. In *CComputer Science Review*, Volume 27, Pages 16-32.

16. Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, Hal Daume III. (2015). Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681-1691.

17. Mohsen Farhadloo and Erik Rolland. (2016). Fundamentals of Sentiment Analysis and Its Applications. In *Researchgate*, DOI: 10.1007/978-3-319-30319-2_1.

18. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404-411.

19. Ricardo Campos, Vitor Mangaravite, Arian Pasquali, Alipio Jorge, Celia Nunes and Adam Jatowt. 2019. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509 (2020) 257-289.

20. Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. In *Journalism Bulletin*, 30(4):415-433.

21. Yelena Mejova. 2009. Sentiment Analysis: An Overview. In *academia*.

22. Yequan Wang, Minlie Huang, Li Zhao and Xiaoyan Zhu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In*Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP)*, pages 606-615.