Text Mining-Sentiment Analysis

by

Guo Zhenyue, Jerry

(1630003017)

A Final Year Project Thesis (COMP4004; 3 Credits)

submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Science (Honours)

in

Computer Science and Technology

at

BNU-HKBU

UNITED INTERNATIONAL COLLEGE

Nov, 2019

DECLARATION

I hereby declare that all the work done in this Project is of my independent effort. I also certify that I have never submitted the idea and product of this Project for academic or employment credits.

Guo Zhenyue, Jerry
(1630003017)

____

Date: ____________________

BNU-HKBU

United International College

Computer Science and Technology Program

We hereby recommend that the Project submitted by Guo Zhenyue, Jerry entitled "Text Mining-Sentiment Analysis" be accepted in partial fulfillment of the requirements for the degree of Bachelor (Honours) of Science in Computer Science and Technology Program.

__________________________ __________________________

Date: __________________ Date: ____________________

ACKNOWLEDGEMENT

I would like to express my great gratitude towards my supervisor, Dr. Rui Meng, Rachel who had given me invaluable advice to this project.

Text Mining-Sentiment Analysis

by

Guo Zhenyue, Jerry

(1630003017)

A Final Year Project Thesis (COMP4004; 3 Credits)

submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Science (Honours)

in

Computer Science and Technology

at

BNU-HKBU

UNITED INTERNATIONAL COLLEGE

Nov, 2019

ABSTRACT

The emergence of social software helps us communicate better with others, improving the efficiency of our message delivery and reducing the high cost of communication. In essence, emotions are attached to the language, and then through the transmission of social software to achieve the purpose of communication. But at the same time, language does not fully express our own emotions. (High, 2014)For example, the emotional message always expressed by multiple dimensions (spoken language, body language, etc.) Lacking of enough dimensions caused noise and misunderstanding in our communication. How to classify and mine this information and identify the emotional features contained in it is the research difficulty in the field of natural language processing. It also led many researchers to carry out related research, and the techniques related to text sentiment analysis came into being. The current analysis of sentiment analysis on social media commentary has limited effect on the sentiment analysis of dialogue directly due to the large differences in language structure between comments and dialogue. At present, mainstream research methods are divided into two types: emotional dictionary and deep learning. This paper uses the tagged real conversation dataset, uses the BERT (Bidirectional Encoder Representation Transformer), developed by Google to conduct deep learning, and builds a chat room based on TCP protocol, and applies the trained script to the chat window. Based on the original model, the final parameters and model packages are optimized by comparing the experimental data and parameter adjustments of different model packages.

Contents

1 Introduction

1.1 Inspiration

Emotion plays an important role in communication and it also affects people in many aspects, such as intelligence, judgment, memory and so on. For individuals, communication's function is to inform each other of their current state through emotions. On the one hand, give feedback on the state of the previous interaction, on the other hand, let others infer the tendency of ourselves. Emotional bandwidth is a definition which refers to the size and dimension of the amount of information you can express. But in the chat on the internet, things get different with the change of bandwidth. For example, without the emoji, stickers and typing status, sometimes we can't express our emotions clearly. In order to help people get a well understanding with text in communication, I hope to design a system that can identify the emotion of user in chat, and provide some suggestion for users who use the system. This system may help a lot for the person who has social disorder.

1.2 Process

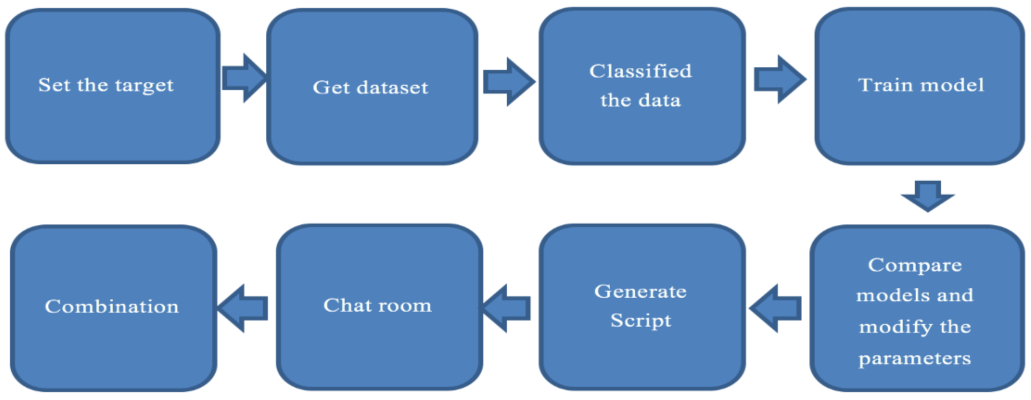


Figure 1 Process of project

1.3 Difficulties and Target

The major difficulties in the whole project is to get the association with the previous content in dialogue. For example, the sunshine in “Sunshine Garden” is different from the sentence “you are my sunshine”. There is another thing that we need to notice is irony. For example, A: “He is 160 cm tall” B: “I think he is 165 cm tall” A: “No matter what the number is, he is really tall”. If we can handle these things well, maybe we can improve our accuracy a lot. The key point is how much previous context we catch each time and how we predict the emotion. Our target is to train the model as perfect as possible and provide the guidance for person to get the emotion of opposite when chatting.

1.4 Some examples

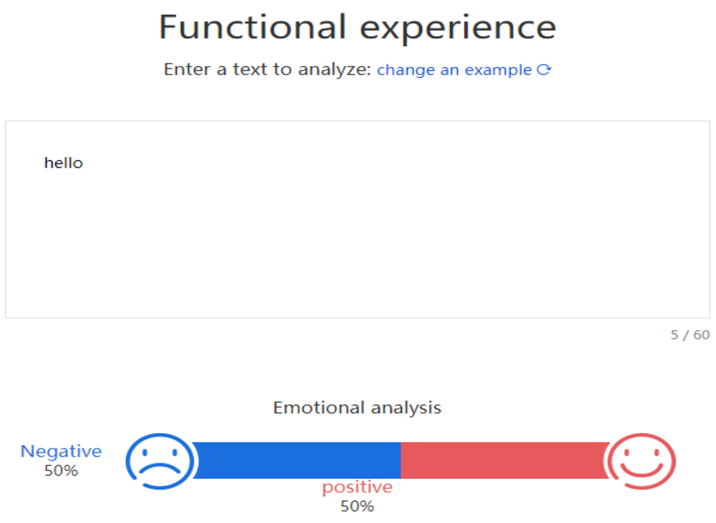
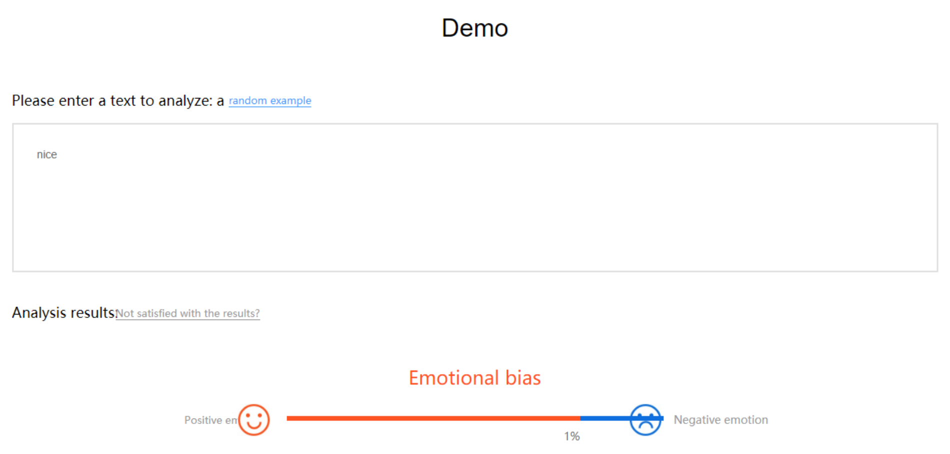


Figure 2 Product of Tencent


Demo

Please enter a text to analyze: a random example

nice

Analysis resultsNot satisfied with the results?

Figure 3 Product of Baidu

These are the same-type products of leading companies in China, which performs quite well. But if we test something special, the result always generated in a wrong way. In our project, we apply it in chat room rather than detect on website.

1.5 Major work

The project is divided into two parts: training model and build a chat room. For training model, the most important step is to select model that fit our dataset. Then build our chat room to provide a platform for detection. The last step is to check the effort of detection and optimize the parameters to help program perform better.

1.5.1 Model selection and application

At present, there are many models in sentiment analysis. We compare the cutting-edged models based on the data set, evaluate their performance, and choose the model that is most suitable for our training.

1.5.2 Chat room based on TCP

TCP is the major protocol of network and most of social software are based on this protocol. Such as, QQ and WeChat, the leading product in China. So, we build the chat room like them and named it as TaChat. The whole project is designed and implemented by MVC model.

1.5.3 Optimization

Considering the performance of model, without testing all parameters cannot prove the current effect is good. We focus on some parameters to adjust model fit for our dataset.

2 Related work

Text sentiment analysis is a multi-disciplinary research field. There are many mature related theories and technologies to use in the fields of natural language processing, machine learning, data mining, and other related resources and sentiment analysis such as corpora and sentiment dictionary. Experimental evaluation platform, these technologies and resources provide well support for sentiment analysis.

2.1 Process of sentiment analysis

It is generally believed that the true text sentiment analysis process refers to text sentiment discrimination and text sentiment intensity evaluation. (Yates, Goharian, & Yee, 2013) Text emotion polarity discrimination is automatic recognition of positive, positive, ambiguous or negative, negative, and derogatory emotions on the subjective text table, while text emotion intensity evaluation refers to quantifying the emotions expressed by subjective text polarity is generally divided into five categories: strong derogation, general derogation, neutrality, general praise, and strong praise.

2.2 Dataset and pre-process

Text preprocessing mainly includes the process of word segmentation, part-of-speech tagging, etc. It is a relatively mature technology related to natural language processing. (Pak & Paroubek, 2011)There are already related software and open platforms in China for natural language researchers. The dataset is like the fuel of the vehicle; you can’t drive it if the tank is empty. So, you are not able to train the model without a dataset. Meanwhile, the dataset should not only large enough to increase the stability of training but also need to be classified into 4 classes. The classes are Angry, Happy, Sad and others. We found some useful data which came from real dialogue in social network. The structure of them are highly adapt to our project, which can help the model perform better.

2.3 Related method

2.3.1 RNN

RNN means recurrent neural network and its feature is quite fit for sentiment analysis in a dialogue. Comparing with a traditional neural network, RNN shares the same parameters rather than using different parameters in different layers. The shortage of RNN is that it can’t compute concurrently, which is a huge waste for computational power. It is convenient because it reduces the parameters we need to know and easy to understand. The diagram below can help us know more about RNN.

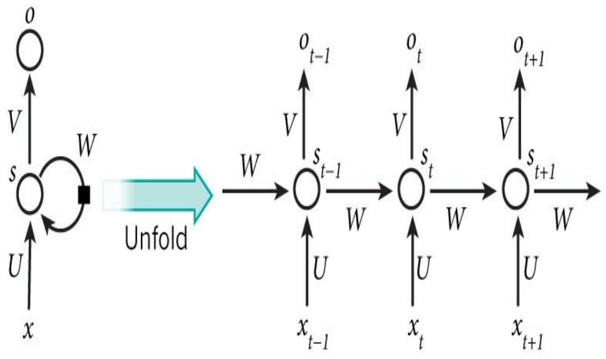


Figure 4 Basic structure of RNN

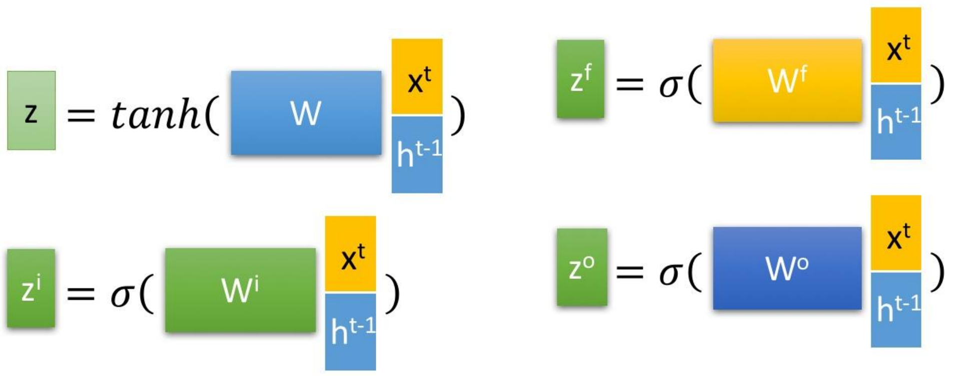
$$S_t = f(Ux_t + Ws_{t-1})$$

Equation 1 Basic function of RNN

x_t means the input at step t. For S_t, it is a hidden state at step t. We can treat it as the memory of the network and it gathers information from all the previous steps. s_t is the result of the previous hidden state's calculation. o_t is the output at s_t. The diagram actually shows the RNN with unfolded and folded shapes. In order to explain more clearly, we can imagine that we have a sentence contains five words. Each layer in RNN will in charge of one word. We don't need to know each word's sentiment; the final output is fine. In a short conclude, RNN captures some information about a sequence which is the main feature of it. (Yi, 2018)However, in some cases we need more contextual information. Think about it, we want to predict the last word of the following sentence "I grew up in France ... I speak fluent French." Which is "French". According to the previous information "I speak fluent" we know that the next word should be a name of language, but what language is it? We want to determine which language must get more information from the earlier sentence "I grew up in France". It is very likely that the distance between the relevant information and the location where the information is needed is very large. Unfortunately, as the distance increases, the RNN becomes unable to connect to relevant information.

2.3.2 LSTM

The full name of Long Short-Term Memory Network (LSTM) is Long Short Term Memory networks, which is a special form of RNN, which is capable of learning long-distance dependencies.



$z = tanh(W \; x^t \; h^{t-1})$
$z^f = \sigma(W^f \; x^t \; h^{t-1})$
$z^i = \sigma(W^i \; x^t \; h^{t-1})$
$z^o = \sigma(W^o \; x^t \; h^{t-1})$

Figure 5 Status of LSTM

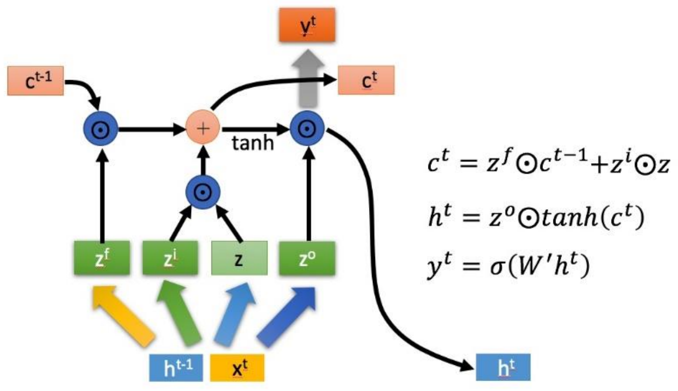


Figure 6 Process of LSTM

$\odot$ is Hadamard Product, that is, the corresponding elements in the operation matrix are multiplied, so the two multiplication matrices are of the same type. (Sundermeyer, Schlüter, & Ney, 2012)

There are three main phases in LSTM:

Forgotten stage

This stage is mainly to selectively forget the input from the previous node. In simple terms, it will forget the unimportant part and remember the important part. Specifically, the calculated z^f (f represents forget) is used as the forgetting gate to control the previous state c^{t-1}, which needs to be left and which needs to be forgotten.

Memory selection stage

This stage selectively "remembers" the input from this stage. It is mainly to select and remember the input x^t . It is important to keep a record of what is important, and less of it. The current input is represented by the z calculated earlier. The selected gate control signal is controlled by z^i (i stands for information). Add the results obtained in the above two steps to get the c^t transmitted to the next state. This is the first formula in the figure above.

Output stage

This phase will determine which outputs will be treated as the current state. It is mainly controlled by z^o. And the c^o obtained in the previous stage is also scaled down (changed by an $\tanh$ （c^t） activation function).

LSTM perform better than simple RNN but it also introduces a lot of content, which leads to more parameters and makes training more difficult.

2.4 Summary

This chapter listed some methods of sentiment analysis and basic procedure of sentiment analysis. Also, introduced to the dataset and pre-processing of it. The content above is part of the theoretical basis of related work.

3 Sentiment Analysis Based on BERT

Just like RNN and LSTM mentioned earlier, BERT is also a model that performs well in sentiment analysis. Unlike them, Bert's performance has been greatly improved. (Devlin, Chang, Lee, & Toutanova, 2018) Therefore, we choose to use this model to complete the detection of dialog emotions. Here we will mainly introduce the architecture and characteristics of the Bert model, while focusing on how BERT plays a role in our dialog sentiment analysis task.

3.1 Theory of BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It is a new language model developed and released by Google at the end of 2018. Pre-trained language models similar to the BERT model, such as question answering, named entity recognition, natural language reasoning, text classification, etc., play an important role in many natural language processing tasks. (Google, 2019) The article next introduces him from several aspects. BERT is a multi-layer bidirectional Transformer encoder based on trimming. It is particularly important to first introduce the architecture of the Transformer. Before elaborating on the Transformer model, let us first introduce Attention.

3.1.1 Attention

Attention mechanism can be regarded as a form of fuzzy memory. Memory consists of the hidden state of the model, and the model chooses to retrieve content from the memory. Before diving into the Attention, let's briefly review the Seq2Seq model. Traditional machine translation is mainly based on the Seq2Seq model. The model is divided into encoding layer and decoding layer, and consists of RNN or RNN variants (LSTM, GRU, etc.). (Attention is All you Need, 2017)The coding vector is the final hidden state generated from the coding part of the model. This vector is designed to encapsulate the information of all input elements to help the decoder make accurate predictions. It is used to serve as the initial hidden state of the model decoder section. The main bottleneck of the Seq2Seq model is the need to compress the entire content of the source sequence into a fixed-size vector. If the text is slightly longer, it is easy to lose some information about the text. To solve this problem, Attention came into being. The mechanism alleviates this problem by having the decoder review the hidden state of the source sequence and then providing its weighted average to the decoder as additional input. Using Attention, as the name implies, the model selects the context that is most suitable for the current node as input during the decoding phase. The figure below shows the mechanism of Attention:

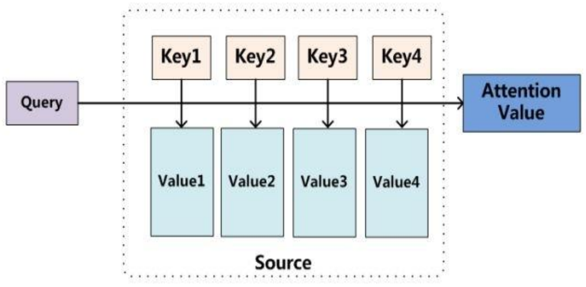


Figure 7 Mechanism of Attention

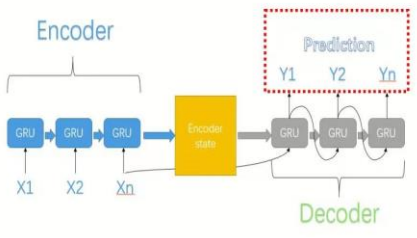
Here, imagine the constituent elements in source sentence as a database composed of <Key, Value> data pairs. At this time, given an element q in a continuous query sequence Query and get the weight coefficient of Value corresponding to each Key by calculating the similarity or correlation between q and each Key. Then perform weighted summation on Value to obtain the final Attention value. So in essence, the Attention mechanism is a weighted summation of the Value values of the elements in the Source, and q and Key are used to calculate the weight coefficient of the corresponding Value. The equation of Attention is:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Equation 2 Equation of Attention

And softmax is a function that takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.

There are two main differences between Attention and the traditional Seq2Seq model. (Gehring, Auli, Grangier, Yarats, & Dauphin , 2017)First, the encoder provides more data to the decoder, and the encoder provides the hidden state of all nodes to the decoder, not just the hidden state of the last node of the encoder. Second, the decoder does not directly take the hidden states provided by all encoders as input, but uses a selection mechanism to select the hidden state that best matches the current position. To this end, it tries to determine which hidden state has the highest correlation with the current node by calculating the score value of each hidden state and performing a softmax calculation of the score, which makes the higher correlation of the hidden state have a larger score value. Too relevant hidden states have smaller score values. It then multiplies each hidden state with its softmax score to amplify the hidden state with a high score and overwhelm the hidden state with a low score. This scoring exercise is done at each iteration time on the decoder side.



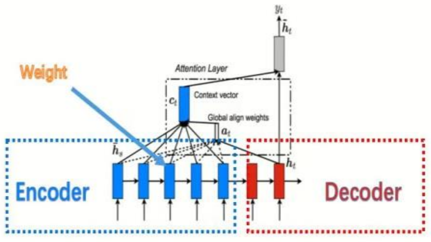
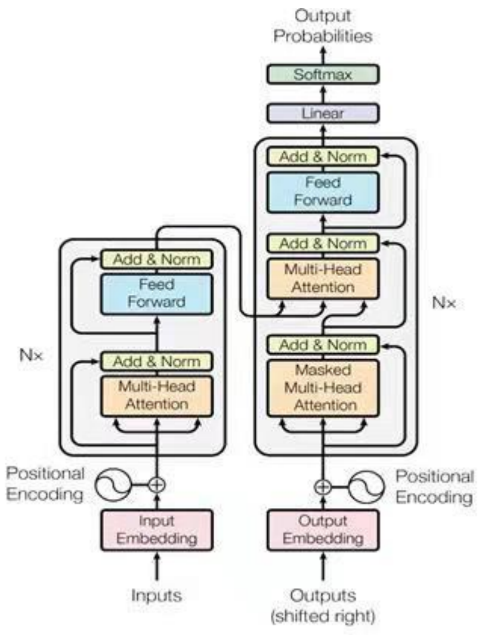


Figure 8 Basic mechanism of Seq2Seq Figure 9 Basic mechanism of Attention

3.1.2 Transformer



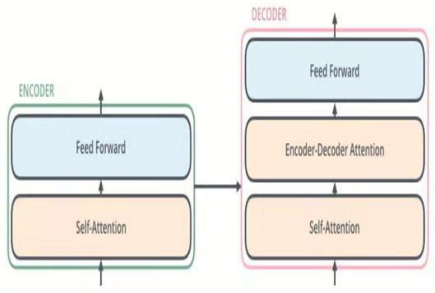


Figure 10 Structure of Transformer Figure 11 Structure of encoder & decoder

Like most Seq2Seq models, the structure of the Transformer is also composed of an encoder and a decoder.

Encoder

Encoder consists of 6 identical layers. The layer refers to the unit on the left side of the figure above. Each encoder consists of two layers, a Multi-Head Self-Attention Mechanism layer and a Fully Connected Feed-Forward network. Each of these sub-layers adds a residual connection and normalization, so the output of the sub-layer can be expressed as:

$$\mathrm{sub\_layer\_output} = \mathrm{LayerNorm}(\mathrm{x} + (\mathrm{SubLayer}(\mathrm{x})))$$

Equation 3 Output of sub-layer of encoder

As for Multi-Head Self-Attention Mechanism layer, its duty is to translate the "understanding" of other words into the processed word. Multi-Head Self-Attention is the updated version of Attention because Query, Key, Value are projected through multiple different linear transformations, and finally the different attention results are stitched together. Here are the equation of Multi-Head Self-Attention. (Devlin, Chang, Lee, & Toutanova, 2018)

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_h)W^o$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Equation 4 Equation of Multi-Head Self-Attention

As for Fully Connected Feed-Forward network, it is to project the refined vector obtained by Multi-Head Attention to a larger space. In that large space, you can more easily extract the required information (using the ReLu activation function), and finally project back to the original space of the token vector.

Decoder

The decoder also contains the two-layer network mentioned by the encoder, but there is also an Attention layer in the middle of the two layers to help the current node obtain the key content that needs Attention.

Positional Encoding

The Transformer adds an additional vector position encoding at the input of the encoder and decoder layers. Its dimensions are the same as the embedded dimensions. This position-encoded value is added to the value of the embedding layer and sent as input to the next layer. There are many position coding options, including learning and repair.

Residual connection and Normalization

In the encoder and decoder, residual connections are used around both sublayers, and then layer normalization is performed. Internal covariate offset refers to the covariate offset that occurs within the neural network, that is, from low to high levels. This is because when the neural network learns and the weights are updated, the output distribution of a specific layer in the network changes. This forces higher layers to adapt to the drift, which slows down learning. Normalization helps to solve the phenomenon that the learning speed slows down due to covariate shifts occurring in the neural network. After normalizing the input in the neural network, there is no need to worry about the large scale of the input features. (Pang & Lee, 2008)

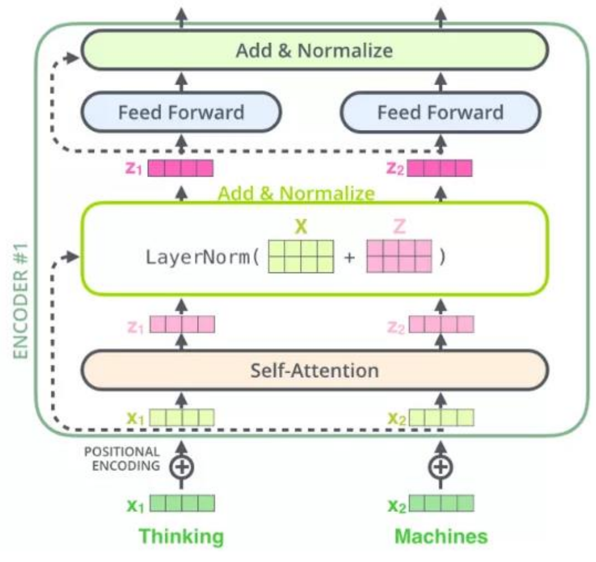


Figure 12 Mechanism of Transformer

3.1.3 Pre-training in BERT

The BERT model pre-training phase includes two unsupervised prediction tasks: masking the language model and predicting the next sentence. When training a BERT model, MLM and NSP are trained simultaneously, the goal is to minimize the combined loss function of the two strategies.

Masked Language Model (MLM)

In order to train the deep bidirectional representation, the input of some percentage (15% in the paper) of the input token is simply randomly masked and then predicted those blocked tokens. (Devlin, Chang, Lee, & Toutanova, 2018)Unlike pre-training of left-to-right language model, MLM goals allow the representation of fused left and right contexts, which makes it possible to pre-train a deep bidirectional Transformer. Although this allows obtaining a bidirectional pre-trained model, the disadvantage is that there is a mismatch between pre-training and fine-tuning, as the [MASK] marker does not appear during fine-tuning. To alleviate this, the author does not always replace the "masked" words with actual [MASK] tags. The training data generator randomly selects 15% of the token positions for prediction. If the i-th token is selected, replace it with (1) the [MASK] token 80% of the time (2) the random token 10% of the time (3) the unchanged i-th token 10% of the time. The BERT model loss function only considers predictions of masked values and ignores predictions of unmasked words. Therefore, the model converges more slowly than the directional model, a feature that is offset by its increased context perception.

Next Sentence Prediction (NSP)

In order to train a model that understands relationship of sentences and semantic relationships between words, BERT also execute the next sentence prediction task, which can be easily generated from any text corpus. Choose some sentences for A and B, where 50% of the data is the situation that B is the next sentence of A, and the remaining 50% of the data is randomly selected in the corpus, and then learn the correlation. The purpose of adding this pre-training is that many natural language processing tasks (such as QA and NLI) need to understand the relationship between two sentences so that the pre-trained model can better adapt to these tasks. To help the model distinguish between the two sentences in training, input is processed as follows before entering the model:

1) Insert [CLS] tag at the beginning of the first sentence, and insert [SEP] tag at the end of each sentence.
2) Add a sentence embedding representing sentence A or sentence B in each tag.
3) Add a positional embed to each marker to indicate its position in the sequence. The concept and implementation of position embedding have been given in the converter paper.

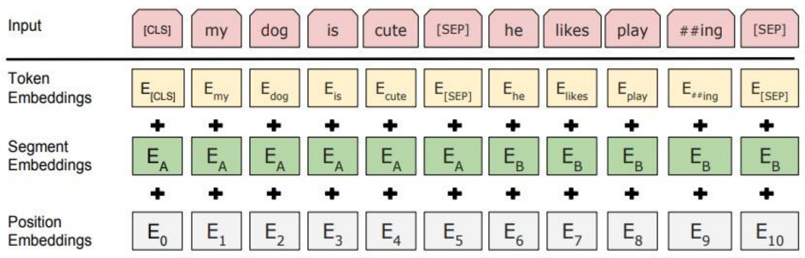

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 13 Process of embedding

To predict whether the second sentence is indeed connected to the first sentence, the following steps need to be performed:

1) The entire input sequence passes through the Transformer model.
2) Use a simple classification layer (a learning matrix of weights and biases) to convert the output of the [CLS] label into a 2×1 vector.
3) Calculate the probability of IsNextSequence using softmax.

3.1.4 Fine tuning in BERT

For each natural language processing task, simply insert task-specific inputs and outputs into the BERT model and fine-tune all the parameters from end to end. At the output, tokens are fed to the output layer for label-level tasks, such as sequence tags or question answers, and [CLS] is fed to the output layer for classification, such as implication or sentiment analysis.

3.1.5 Summary

The innovation of BERT is that it uses a bidirectional Transformer for the language model, which takes better advantage of the Attention mechanism and greatly improves the accuracy of natural language processing tasks. (Lee & Hsiang, 2019)

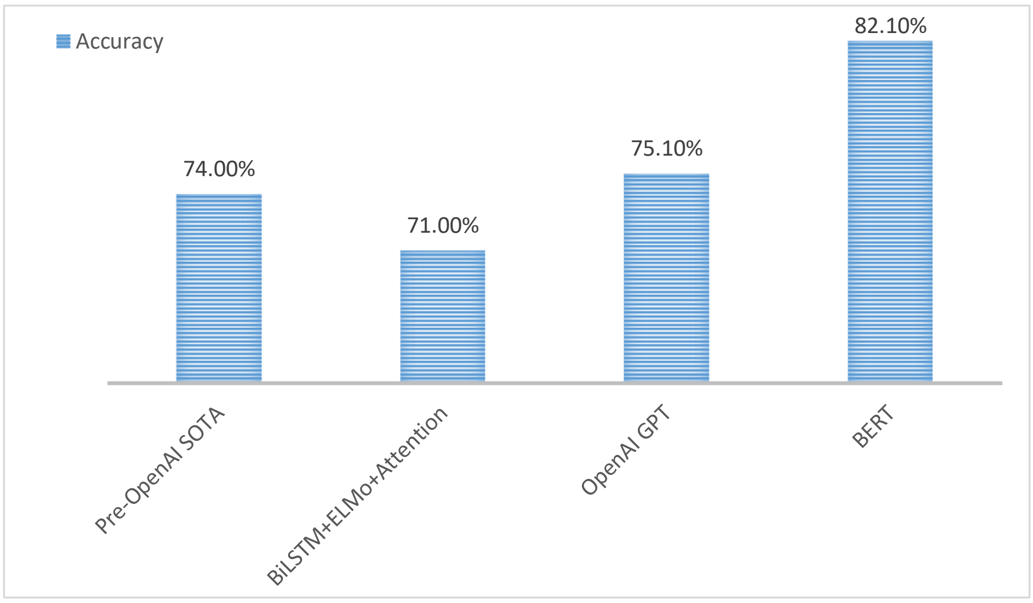
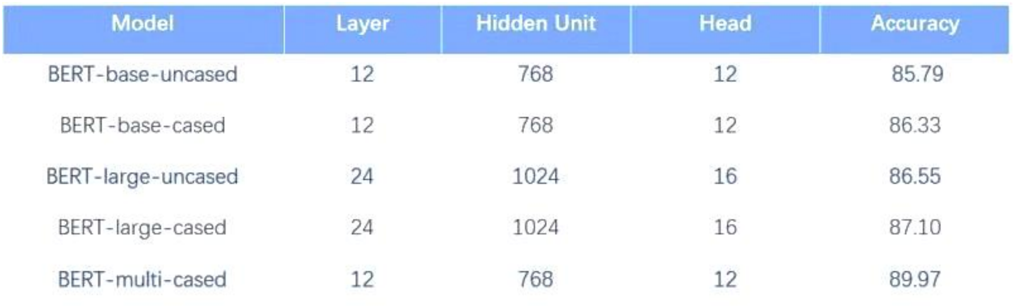


Figure 14 Comparing results of different method in our experiments

3.2 Application on sentiment analysis

As mentioned previously, Fine tuning can apply in sentiment analysis and the method is loading the pre-trained BERT model. Actually, the process is transfer the data set of specific domain tasks to the model, continue to back-propagate training on the network, and constantly adjust the weight of the original model to obtain a suitable for new task-specific models.

3.2.1 Pre-trained model



Model	Layer	Hidden Unit	Head	Accuracy
BERT-base-uncased	12	768	12	85.79
BERT-base-cased	12	768	12	86.33
BERT-large-uncased	24	1024	16	86.55
BERT-large-cased	24	1024	16	87.10
BERT-multi-cased	12	768	12	89.97

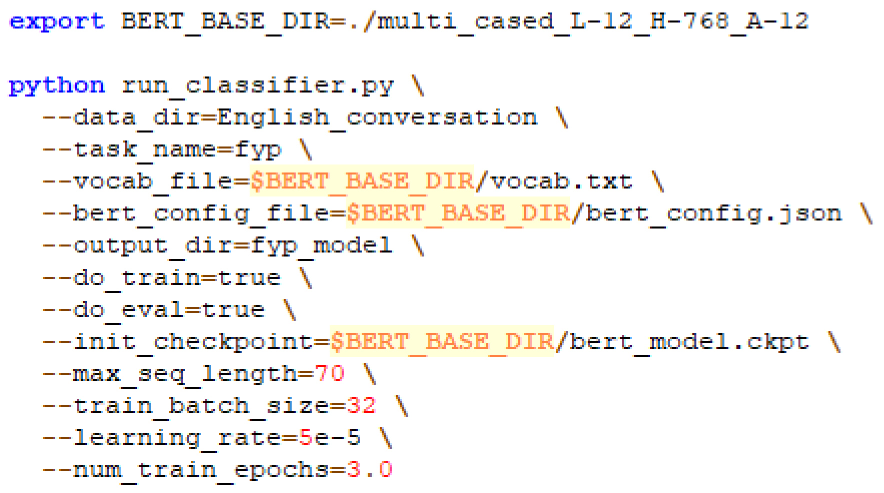
Figure 15 Comparing results of different model of BERT

Google has fulfilled the requests of developers from various countries, and has released pre-trained models of BERT for most languages. Considering the diversity of the language of chat, we have compared performance of multiple models on our own dataset, and select BERT-multi-cased as our model.

3.2.2 Build a processor

For a model that needs to perform the entire process of training, cross-validation, and testing, the custom processor needs to inherit the Data Processor, and overload the **get_labels** to get labels and **get_train_examples**, **get_dev_examples**, and **get_test_examples** functions to get a single input. It will be called in the **FLAGS.do_train**, **FLAGS.do_eval** and **FLAGS.do_predict** stages of the main function. The contents of these three functions are almost the same, the only difference is that you need to specify the address of each read file. (AI 科技大本营, 2018)

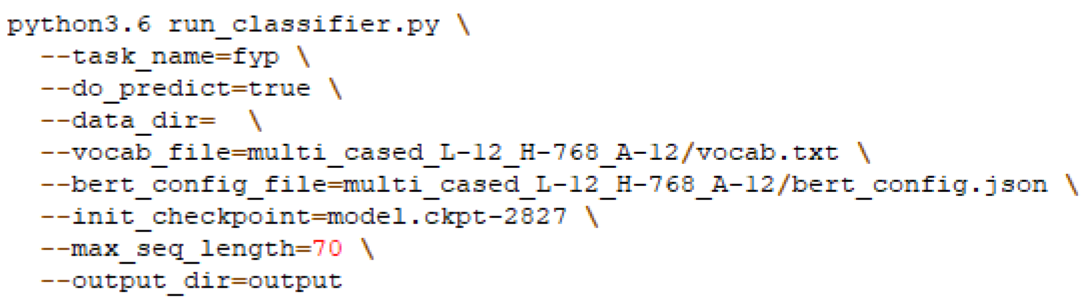
3.2.3 Training and predicting



```
export BERT_BASE_DIR=./multi_cased_L-12_H-768_A-12

python run_classifier.py \
  --data_dir=English_conversation \
  --task_name=fyp \
  --vocab_file=$BERT_BASE_DIR/vocab.txt \
  --bert_config_file=$BERT_BASE_DIR/bert_config.json \
  --output_dir=fyp_model \
  --do_train=true \
  --do_eval=true \
  --init_checkpoint=$BERT_BASE_DIR/bert_model.ckpt \
  --max_seq_length=70 \
  --train_batch_size=32 \
  --learning_rate=5e-5 \
  --num_train_epochs=3.0
```

Figure 16 Training script of model

In training process, we need to define some parameters like figure above.

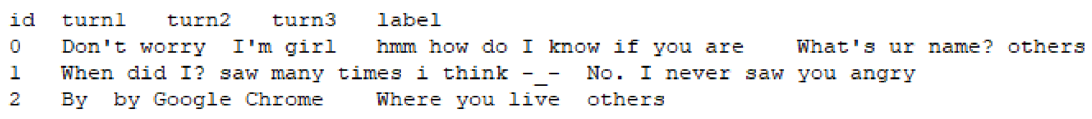


```
python3.6 run_classifier.py \
  --task_name=fyp \
  --do_predict=true \
  --data_dir=  \
  --vocab_file=multi_cased_L-12_H-768_A-12/vocab.txt \
  --bert_config_file=multi_cased_L-12_H-768_A-12/bert_config.json \
  --init_checkpoint=model.ckpt-2827 \
  --max_seq_length=70 \
  --output_dir=output
```

Figure 17 Predicting script of model

Same as the training method, script of prediction also need to transform some parameters into it. The major difference is that the model here **init_checkpoint=model.ckpt-2827** is the model after fine tuning on our dataset**.** All the result will be automatically record in a .tsv file and save in output directory.

3.2.4 Optimization



```
id  turn1   turn2   turn3   label
0   Don't worry  I'm girl   hmm how do I know if you are    What's ur name? others
1   When did I? saw many times i think -_-  No. I never saw you angry
2   By  by Google Chrome    Where you live  others
```

Figure 18 Part of dataset

According to the suggestion of Google official, we set the learning rate as $5e-5$, max length of a sentence as 70, **num_train_epoch** as 3 (train repeatly 3 times).

We mainly focus on the combination of sentence. Each record in our dataset consists of 3 sentence but BERT can only accept 2 of them. (Google, 2019) So we test the combination of them.

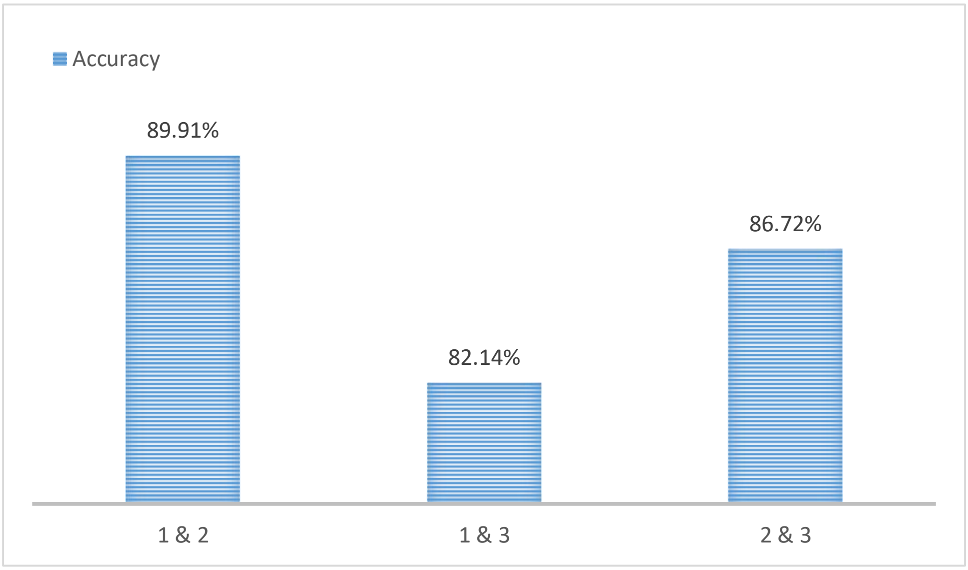


Figure 19 Comparing result of different combination

4 Design of TaChat

This chapter will introduce the technology we use in the process of designing and developing TaChat and explain the specific functions and implementation mechanisms.

4.1 TCP Protocol

The basic principle of the chat room is using client / server TCP architecture, the client sends a message to the server, and the server forwards the message to all clients. We chose TCP for development because of its reliable and stable characteristics. Before transmitting data, TCP will establish a three-way handshake to establish a connection, and during data transmission, there are mechanisms for confirmation, windowing, retransmission, and congestion control. After the data transmission is completed, the connection will be disconnected to save system resources. These advantages can make our chat room faster and more reliable.

4.2 MVC Model

The MVC pattern (Model-View-Controller) is a software architecture pattern in software engineering. It divides the software system into three basic parts: Model, View, and Controller. (Lucassen & Maes, 2000)

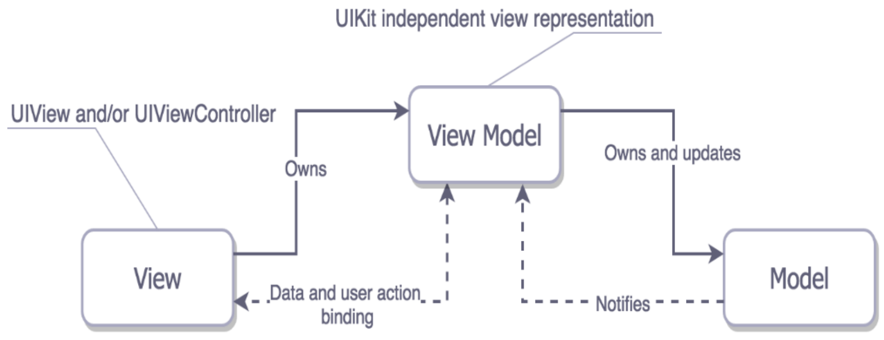

UIKit independent view representation
UIView and/or UIViewController
View Model
Owns
Owns and updates
View
Model
Data and user action binding
Notifies

Figure 20 MVC model

We chose to establish our project based on this model because it has these advantages:

1) Low coupling. The view layer is separated from the business layer. This allows the view layer code to be changed without recompiling the model and controller code. Similarly, changes in an application's business processes or business rules need only change the MVC model layer. Because we can't do development at the same time, we can reduce a lot of repetitive labor and avoid the high time cost caused by mistakes in the case of separate work.
2) High reusability and applicability. As technology continues to advance, applications need to be accessed in more and more ways. The MVC pattern allows you to use different styles of views to access the same

server-side code. It includes any web (HTTP) browser or wireless browser (WAP). Because the data returned by the model is not formatted, the same components can be used by different interfaces. This leaves a good technical architecture for future functional expansion.

3) Quick deployment.
4) Maintainability. Separating the view layer and the business logic layer makes the application easier to maintain and modify.
5) Conducive to software engineering management. Because different layers perform their duties, different applications at each layer have certain same characteristics, which is conducive to managing program code through engineering and tooling.

4.3 Function

Our chat room is modeled after the WeChat chat function. The following are the features of our chat room:

1) Login and logout as most of social application.
2) Support chat from multiple people to multiple people at the same time
3) Finish sentiment analysis and suggestions on users' chat content if user want.
4) Save chat history in local.

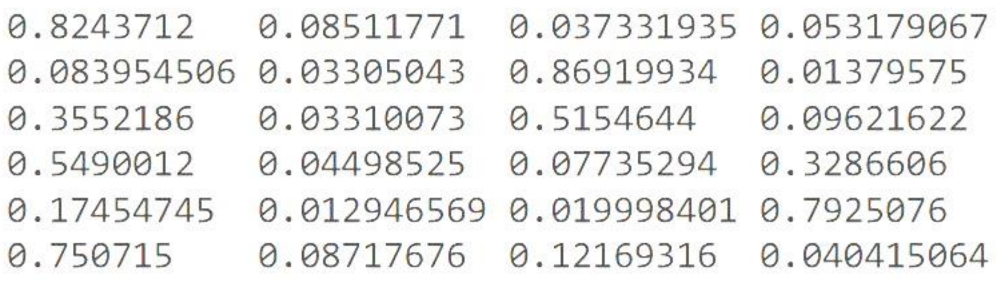
4.4 Application of script on TaChat

This section mainly describes how to load the obtained script into the chat room and display the test examples.

4.4.1 Working mechanism

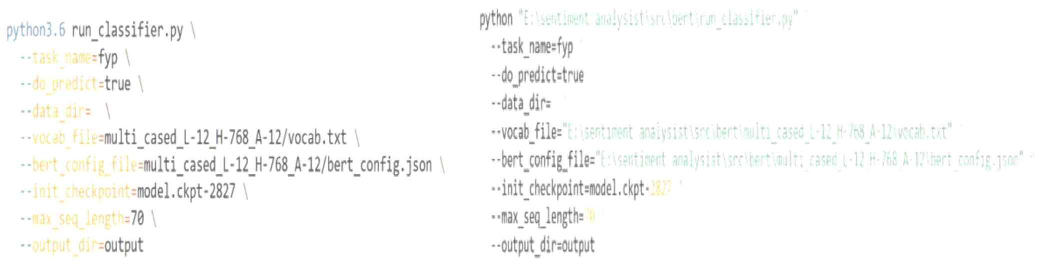
As for how to combine the script with the chat room, my idea is to extract the chat content as a whole, then pass them into the script for prediction, and finally return the result to the chat room to make suggestions.

First, we extract the content of the chatter into a .tsv file, which is determined based on the model definition provided by Google. Next, run the predict.sh script to make predictions. However, because the windows system cannot run the file in .sh format, we convert it as a bat file to ensure that the prediction can proceed smoothly. (Chen, 2013) The time spent in the entire prediction process is determined by the computing power and data size, which is mainly affected by the computing power in this case. Then we can get the predicted result. Each column represents one of the 4 labels and the row means one sentence. The prediction result here is the probability of the four given labels (other, angry, sad, and happy). The greater the probability, the more the result is biased towards the corresponding label. The sum of all the probabilities is 1. The predicted result is stored in the .tsv file. All the steps based on BERT are finished and wait for the final step. Finally, in the chat room, we make a simple comparison among probabilities, return the results directly in the suggestion, and tell the user's emotional state through four types of emoticons and text prompts.



```
0.8243712   0.08511771  0.037331935 0.053179067
0.083954506 0.03305043  0.86919934  0.01379575
0.3552186   0.03310073  0.5154644   0.09621622
0.5490012   0.04498525  0.07735294  0.3286606
0.17454745  0.012946569 0.019998401 0.7925076
0.750715    0.08717676  0.12169316  0.040415064
```

Figure 21 Output of prediction



```
python3.6 run_classifier.py \
  --task_name=fyp \
  --do_predict=true \
  --data_dir=  \
  --vocab_file=multi_cased_L-12_H-768_A-12/vocab.txt \
  --bert_config_file=multi_cased_L-12_H-768_A-12/bert_config.json \
  --init_checkpoint=model.ckpt-2827 \
  --max_seq_length=70 \
  --output_dir=output
```

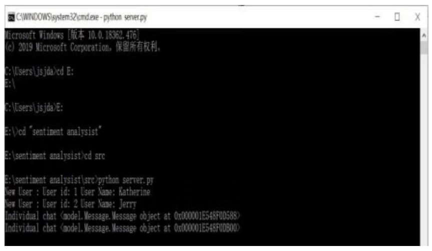
```
python "E:\sentiment analysist\src\bert\run_classifier.py" ^
  --task_name=fyp ^
  --do_predict=true
  --data_dir= ^
  --vocab_file="E:\sentiment analysist\src\bert\multi_cased_L-12_H-768_A-12\vocab.txt" ^
  --bert_config_file="E:\sentiment analysist\src\bert\multi_cased_L-12_H-768_A-12\bert_config.json" ^
  --init_checkpoint=model.ckpt-2827 ^
  --max_seq_length=70 ^
  --output_dir=output
```

Figure 22 Convert predict.sh to predict.bat

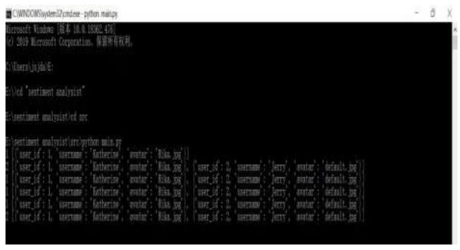
4.4.2 Outcome

This part is for introduction of outcome of software and display whole process. The test data came from the real dialogue has just happened in my daily life.

First, start running the server and clients.



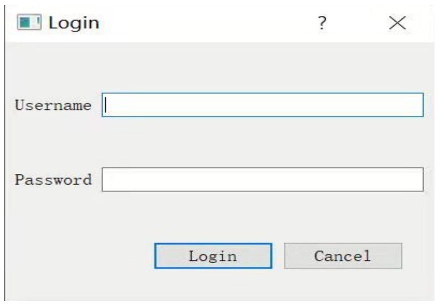
```
C:\WINDOWS\system32\cmd.exe - python server.py
Microsoft Windows [版本 10.0.18362.476]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\jsjda>cd E:
E:\

C:\Users\jsjda>E:

E:\>cd "sentiment analysist"

E:\sentiment analysist>cd src

E:\sentiment analysist\src>python server.py
New User : User id: 1 User Name: Katherine
New User : User id: 2 User Name: Jerry
Individual chat <model.Message.Message object at 0x000001E548F0D588>
Individual chat <model.Message.Message object at 0x000001E548F0D800>
```

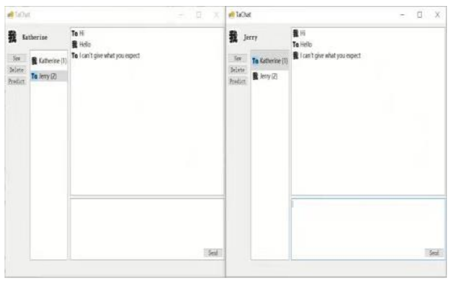


```
C:\WINDOWS\system32\cmd.exe - python main.py
Microsoft Windows [版本 10.0.18362.476]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\jsjda>E:

E:\>cd "sentiment analysist"

E:\sentiment analysist>cd src

E:\sentiment analysist\src>python main.py
1 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}]
2 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}, {'user_id': 2, 'username': 'Jerry', 'avatar': 'default.jpg'}]
2 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}, {'user_id': 2, 'username': 'Jerry', 'avatar': 'default.jpg'}]
2 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}, {'user_id': 2, 'username': 'Jerry', 'avatar': 'default.jpg'}]
2 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}, {'user_id': 2, 'username': 'Jerry', 'avatar': 'default.jpg'}]
2 [{'user_id': 1, 'username': 'Katherine', 'avatar': 'Rika.jpg'}, {'user_id': 2, 'username': 'Jerry', 'avatar': 'default.jpg'}]
```

Figure 23 Server of TaChat

Figure 24 Client of TaChat





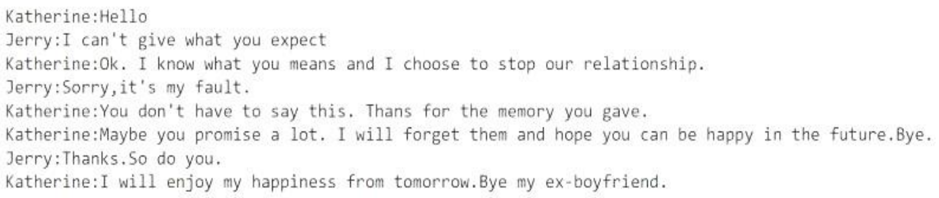
```
To Hi
Hello
To I can't give what you expect
```

```
Hi
To Hello
I can't give what you expect
```

Figure 25 Login page of users

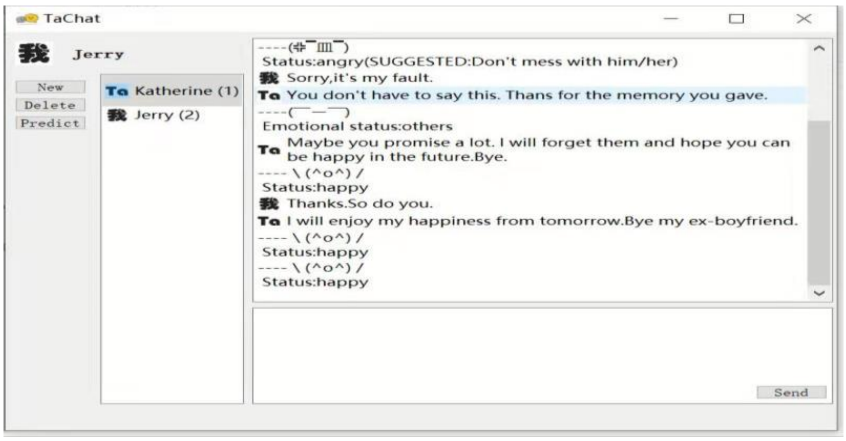
Figure 26 User Interface

After login, they start chatting. Here are the content of dialogue and the figure below display the prediction of result of them.



```
Katherine:Hello
Jerry:I can't give what you expect
Katherine:Ok. I know what you means and I choose to stop our relationship.
Jerry:Sorry,it's my fault.
Katherine:You don't have to say this. Thans for the memory you gave.
Katherine:Maybe you promise a lot. I will forget them and hope you can be happy in the future.Bye.
Jerry:Thanks.So do you.
Katherine:I will enjoy my happiness from tomorrow.Bye my ex-boyfriend.
```

Figure 27 Content of dialogue



```
----(卝¯皿¯)
Status:angry(SUGGESTED:Don't mess with him/her)
我 Sorry,it's my fault.
Ta You don't have to say this. Thans for the memory you gave.
----(¯—¯)
Emotional status:others
Ta Maybe you promise a lot. I will forget them and hope you can
   be happy in the future.Bye.
----\(^o^)/
Status:happy
我 Thanks.So do you.
Ta I will enjoy my happiness from tomorrow.Bye my ex-boyfriend.
----\(^o^)/
Status:happy
----\(^o^)/
Status:happy
```

Figure 28 Prediction

As the demonstration displaying, it can generally predict the emotion of human beings and give a suggestion for user.

5 Conclusion and Future work

Sentiment analysis has always been a hot topic in natural language processing. Currently, there are many sentiment analysis projects for social website comments and short texts. However, there is still relatively little research on sentiment analysis in dialogues. This article uses Google's open source deep learning framework to train existing datasets, and at the same time obtains script files and completes chat software with sentiment analysis suggestions in combination with chat software.

5.1 Conclusion

In the process of building a chat room with sentiment analysis function, this article first completed the crawling and collection of the data set, selected the daily conversation dataset with four tags, and then chose the neural network framework and compared each model's performance on dataset. Such as, Pre-OpenAI SOTA, BiLSTM+ELMo+Attention and OpenAI GPT. (Devlin, Chang, Lee, & Toutanova, 2018) After determining the choice of BERT, we adjusted and optimized its parameters (learning rate and maximum sentence length). At the same time, the model packages of different languages of BERT were compared and BERT-multi-cased one, which has 12 layers, 768 hidden unit and 12 head, became the first choice. As for dataset, we not only removed some noise in it, but also compared the performance of different sentence combinations to select the currently used combination. The combination of sentence 1 and sentence 2 has better performance in experiment. These are the result of our project and it perform well on prediction in chat room. However, there are still many areas that need to be improved and optimized throughout the project.

1) The size of the model is too large to be trained on ordinary machines, and our training process needs a good server to work properly. Although a lot of training can greatly improve the performance of our model, it consumes too much resources and relies too much on computing power.
2) The size of the dataset is not large enough, and the dialogue information contained in it is still in the relatively basic stage of daily life. As a result, the model's turn sensitivity to complex sentences and emotions needs to be improved and enhanced. At the same time, the four tags are still too simple for human complex emotions, and the polarity of emotions often have different results under different conditions, and the training sample of three sentences to form a record is not perfect in complex mission.

3) The development of the chat room based on the windows system has led to incompatibility with the scripts developed by the Linux system, which can only greatly sacrifice the software's response speed. It cannot be placed on a network server to run, and it has great requirements on the computing power of the local computer.

5.2 Future work

In view of the existing problems, we look forward to improvements in the subsequent development, mainly in the function of the chat room and the performance of the software.

5.2.1 Function

Functionally, in addition to optimizing the accuracy of predictions, we also hope to add the ability to automatically reply to messages. This feature is aimed at scenarios where users do not know how to reply to messages. It can provide rich chat speech, replace the user's reply when appropriate, and improve the user experience.

5.2.2 Performance

From the aspect of performance, what should be solved is the problem of slow detection and unresponsiveness to some emotional context scenes. In the future, we will use a new chat room to solve the problem of multithreaded blocking, and then develop based on the Linux system, load the entire script into the chat room, reduce the steps of reading and transmitting. Meanwhile, improve the response speed, which means the part that requires a lot of computing power is moved to the server, speeding up and reducing the use of local computer CPUs and GPUs.

In addition, optimize the model and continue to adjust the parameters to reduce the time required from the prediction process.

6 References

[1] AI 科技大本营. (2018, 12 28). Retrieved from Tencent Cloud: https://cloud.tencent.com/developer/article/1373807

[2] Chen, X. (2013, 8 26). Retrieved from Microsoft Dev Center: https://social.msdn.microsoft.com/Forums/windowsdesktop/en-US/bb6dfef1-4e05-4461-a146-6c8555405d99/convert-sh-into-bat?forum=windowsgeneraldevelopmentissues

[3] Cun, Y. L., ottou, L., & Bengio, Y. (1997). Reading checks with multilayer graph transformer networks. International Conference on Acoustics, Speech, and Signal Processing. Munich: IEEE.

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[5] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin , Y. N. (2017). Convolutional sequence to sequence learning. ICML'17 Proceedings of the 34th International Conference on Machine Learning, (pp. 1243-1252). Sydney.

[6] Google. (2019, 5 31). Retrieved from GitHub: https://github.com/google-research/bert

[7] High, A. C. (2014). Misery rarely gets company: The influence of emotional bandwidth on supportive communication on Facebook. Computers in Human Behavior, 79-88.

[8] Lee, J.-S., & Hsiang, J. (2019). Patent Claim Generation by Fine-Tuning OpenAI GPT-2.

[9] Lucassen, J. M., & Maes, S. H. (2000). MVC (model-view-controller) based multi-modal authoring tool and development environment.

[10] Pak, A., & Paroubek, P. (2011). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. 30-38.

[11] Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends® in Information.

[12] Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. INTERSPEECH , 194-197.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is All you Need.

[14] Yates, A., Goharian, N., & Yee, W. (2013). Semi-supervised probabilistic sentiment analysis: Merging labeled sentences with unlabeled reviews to identify sentiment. Proceedings of the American Society for Information Science and Technology, 1-10.

[15] Yi, Z. (2018, 12 14). Retrieved from Jiqizhixin: https://www.jiqizhixin.com/articles/2018-12-14-4