Sentiment Analysis of Chat Application

Swanand Joshi
Department of Information and Technology
Pune Institute of Computer and Technology
Pune, India

Amey Ruikar
Department of Computer Science
Pune Institute of Computer and Technology
Pune, India

***Abstract*— Sentiment analysis is one of the most recent applications of natural language processing. With the advent of mobile chat applications, sentiment analysis can be done on the textual data that is generated by the application. The proposed sentiment analysis system consists of two phase – Recognition Phase and Distribution Phase. Recognition phase uses as set of features to perform sentiment classification. Logistic regression algorithm is employed to train the classifiers. The training set required for system Recognition comprises of the text messages sent by the user. In the distribution phase, the application server sends out the updated contact lists of all other nodes with the latest evaluated sentiment .This second phase of the system uses PUSH Technology to distribute the result of the sentiment analysis over the network.**

Keywords—sentiment analysis; nlp; machine Recognition; PUSH Technology

I. Introduction

Chat applications generate large amounts of data every day. Every user sends a lot of textual data over the network. The proposed sentiment analysis system uses this data to learn and predict the sentiment or the mood of the user at that particular time. Human being can correctly predict the mood of the user by reading his/her chat messages. To enable a system to do a similar task we need to convert this textual data into features which a machine can understand. Once the features are selected the system classifiers are trained using a training set.

In the second phase of the system, the recently evaluated sentiment of a particular user has to be updated at two places. First is the application server and second, on all the mobile devices that have this particular user as a contact or on the messaging list. The later part has been designed using PUSH technology. PUSH technology has definite advantages over the more widely used PULL technology. The application logic will determine which user’s contact lists have to be refreshed and accordingly a PUSH will be executed towards the device. The application running on the device will process the incoming message following which the status and sentiment will be refreshed.

In the subsequent topic, the overall system architecture is described.

II. System Architecture

The sentiment analysis system comprises of two phases. Recognition and distribution phase. Architecture of the system is described according to the phases as follows.

A. Recognition Phase

The Recognition phase of the system is implemented on every single device. Every user has a device from which he/she uses the chat application. Ex. Mobile phones, Tablets. The Recognition phase module runs after a certain period of time to predict the sentiment of the user. It transforms the textual data into the selected features and using logistic regression, recognizes the emotion. The result of the analysis is sent to the application server for further distribution.

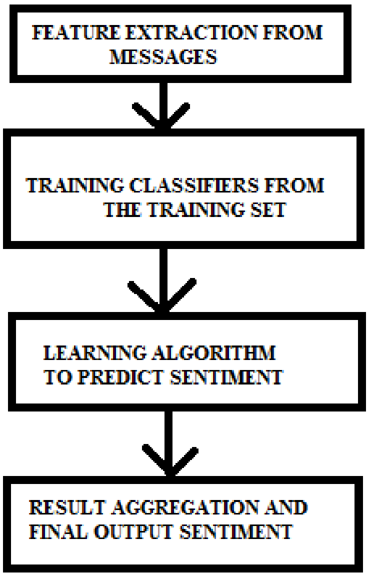
FEATURE EXTRACTION FROM MESSAGES
TRAINING CLASSIFIERS FROM THE TRAINING SET
LEARNING ALGORITHM TO PREDICT SENTIMENT
RESULT AGGREGATION AND FINAL OUTPUT SENTIMENT

Fig. 1 System Architecture for Recognition Phase

B. Distribution Phase

The distribution phase of the system has three components. First is the device on which the Recognition phase will be running. Second is the application server through which all the interactions take place. Third is the device whose status or sentiments list has to be updated. The following diagram gives an overview of the sub-system.

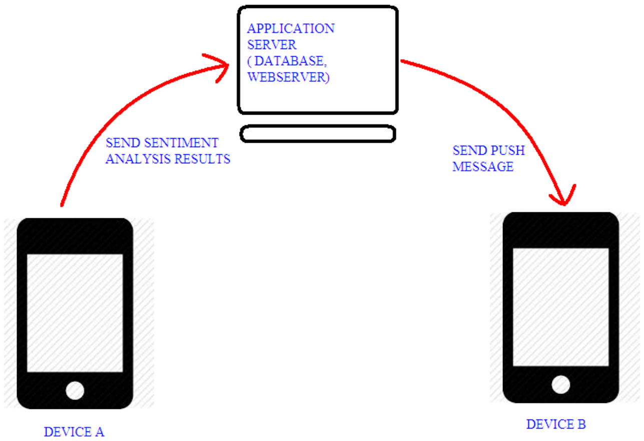
APPLICATION SERVER (DATABASE, WEBSERVER)
SEND SENTIMENT ANALYSIS RESULTS
SEND PUSH MESSAGE
DEVICE A
DEVICE B

Fig.2 System Architecture for Distribution Phase

Device A on the left is a mobile device like a smart- phone or a tablet powered by android or IOS operating systems. The application running on this device will determine the sentiment as mentioned in the recognition phase and forward that information to the main application servers. Further, the logic embedded on the application server will determine to whom all the update should be sent to. The third device, device B, has device A as its contact and as a result it is a recipient of the PUSH message from the application server. The application (android or IOS app) will be installed on device B too. The application on device B will process the incoming message to update the sentiment of that respective contact. These events will also take place from device B to device A.

III. System Description

The proposed system is described according to the two phases involved. The description includes the functionality of the system.

A. *Recognition Phase*

In this phase the actual sentiment analysis is done. The sentiment of the user can be predicted using various ways. In a chat application the most significant factor is the textual data that the user generates. The text messages can reveal a lot about the emotion of the user. Though the context of the messages is the most significant feature, other attributes like emoticons usage, typing speed, length of the messages also contribute in the recognition. The proposed system converts these details into features. There are many algorithms to perform recognition task but this system uses multi-class logistic regression. The sentiment analysis is classified as follows.

Sentiments are measured on two different scales - Emotion and Intensity. The system classifies emotion meter in five classes – Very Happy, Happy, Neutral, Sad and Very Sad. The intensity meter is classified similarly – Very Excited, Excited, Neutral, Muted and Very Muted. The output of the system will be the combination of the two scales. The sentiment will be predicted as a combination of the two scales. This module will run after every time period and calculate the sentiment by analyzing all the text messages sent in this period on every single chat. To find the aggregate sentiment result another algorithm is employed which will be discussed further. This final sentiment result will then be passed to the Distribution Phase as the sentiment of the user in the given time period.

1) Features Selection and Extraction:

The sentiment analysis system requires features to perform sentiment recognition. The features are extracted from the text messages sent by the user. The analysis requires two different classification results – Emotion and Intensity. The feature sets for Emotion and Intensity are different and are briefly explained as follows

- *Vocabulary List*

  The text messages sent by the user are pre-processed and converted into word tokens. The system consists of a vocabulary which includes commonly used positive and negative words such as good, great, poor bad etc. Processed text messages are checked against this vocabulary and found words are converted into features for emotion recognition.

- *Emoticon List*

  Similar to the vocabulary list, the emoticon list includes commonly used emoticons like happy or sad faces. The emoticons present in the text messages are checked against this emoticon list and found emoticons are converted into features for emotion recognition.

- *Status*

  The user status is also used to determine his/her emotion. The status text can be similarly transformed into feature vectors using vocabulary and emoticon lists. The impact of the status gradually lessens in intensity. To achieve this, the status impact factor is calculated and stored as feature. The value of this impact factor gradually decreases with time.

- *Speed*

  The system calculates user’s typing speed and converts into a feature for intensity recognition. A very high typing speed indicates Very Excited user and a very low speed indicates a Very Muted or least intense user.

- *Length*

  In intensity recognition, length of the messages attributes to a certain extent. Message length is high when user is generally excited and the length is condensed when user is very less intense. The system takes lengths of all the messages in the chat and converts into a feature for intensity recognition.

2) Training Classifiers

The recognition phase next undergoes training. A lot of training examples with their emotion and intensity values are fed to the system. The more the training set the better the system learns. The classifiers of the system are trained in this phase and are now ready for actual prediction.

To train the classifiers, logistic regression algorithm is used with a minimizing cost function. To find the minimum cost function for the classifiers, any minimization algorithm can be used, for ex. Gradient Descent.

As the training data provides the correct sentiment analysis of Emotion and Intensity of the training examples, using large datasets will ensure more accurate sentiment analysis.

3) Aggregation of Results

Once the classifiers are trained, actual input can be given to the system. In a particular time span, user can be active on number of chats. The task now remains with the system is to aggregate the individual results into a final and overall sentiment analysis for the user.

The system will predict Emotion and Intensity for every single chat. Now all these results need to be aggregated into a single value of sentiment.

To do this another simple algorithm is employed, where for every n chats, if more than half of the values converge on a single class then the final sentiment is classified as the same class. Otherwise, weights are assigned to each of the class so that the mean calculated gives the accurate sentiment analysis. If the mean value lies between the two classes, a progress bar is shown from the lower value sentiment to the higher value sentiment for example if the mean value is 2.4 and Happy class weight is 1 and Very Happy class weight is 3 then the meter is calibrated accordingly.

These aggregated values of Emotion and Intensity conclude the sentiment analysis of the user in that time period. This analysis is now passed on to the Distribution phase for further processing. The subsequent chapter explains the working of Distribution Phase.

B. Distribution Phase

Both Google (on Android) and Apple (on IOS) have services that let application servers perform PUSH operation. These PUSH messages will be handled by the respective app towards which the message was intended.

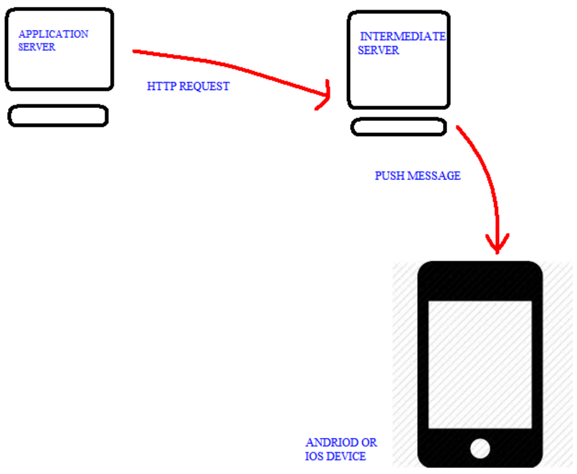

Fig.3 Intermediate server for PUSH services

As we can see in the above diagram, an extra node has been added in the system architecture diagram. This extra node is an intermediate server which actually carries out the PUSH operation. Google names this server as GCM (Google Cloud Messaging) and Apple's version is called the APNS (Apple Push Notification Service) server. The working of this intermediate server on both the platforms is the same. The Intermediate servers have to be configured (keys) before they can start sending Push messages. Also a project number needs to be generated, which is used in the app on individual devices. The flow of events is described further.

First, when the app is installed on the device, the app sends a request to the intermediate server and gets a unique registration ID. Now, this registration ID has to be shared with the application server as well. The registration ID will be stored along with the user name on the application server. These IDs will be used while sending out the messages. After the registration process is done the app will receive Push messages even if it is not in the main memory.

Second, when the application server finds a need to send a message to a particular device it has to only get its corresponding ID. A packet is made with the project number the various keys and the actual data (sentiment) to be sent to the intermediate server. From here on, it is the job of the intermediate server to send PUSH messages to individual devices. The intermediate server also returns the status code for each individual device. Retransmission can take place if the message didn't reach the device first time.

Group Messages can also be sent using these services. Hence if a particular person's sentiment is changed, it can be sent to all people having his contact in one go.

IV. ADVANTAGES

At present, no chat application provides the feature of sentiment analysis. One of the major advantages of the proposed system is that it does not require any user intervention as it runs in the background. It learns user sentiment only from text messages which the user sends. The system can determine the user sentiment – emotion and intensity which will help other contacts to chat with the user.

Using PUSH for the above application has various advantages. First, having an app that continuously syncs data from its respective servers, consumes bandwidth. This also has a negative impact on the battery. Moreover it is difficult to keep an application in-memory; the operating system might decide to kill a few apps depending on the requests for memory. PUSH technology resolves these issues.

V. CONCLUSIONS AND FUTURE SCOPE

The proposed system gives sentiment analysis from the textual data generated by the chat application. The features required for this analysis are extracted simply from the chat messages the user sends and thus requires no additional attributes. The distribution of the analysis takes place using

PUSH technology which tries to maximize efficiency by reducing memory load and network traffic.

As mentioned before, none of the current chat applications include sentiment analysis. The proposed system can be built on top of any chat application such as OpenWhatsapp or an entirely new chat application can be developed.

REFERENCES

[1] Igor Bisio, Alessandro D, Fabio L, Mario M, and Andrea(2013).Gender driven emotion recognition through speech for ambient intelligence application.*IEEE transaction 2013*.

[2] R Prabowo, Sentiment analysis: A combined approach M Thelwall, Journal of Informetrics, 2009 - Elsevier

[3] https://class.coursera.org/ml-005

[4] http://developer.android.com/google/gcm/index.html

[5] https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html