© 2020 IJSRST | Volume 5 | Issue 7 | Print ISSN : 2395-6011 | Online ISSN: 2395-602X

4th National Conference on Advances in Engineering and Applied Science
Organized by : Anjuman College of Engineering and Technology (ACET) Nagpur,
Maharashtra, India, In association with
International Journal of Scientific Research in Science and Technology

Sentiment and Text Analysis

Prof. Kamlesh Kelwade, Zeba Sheikh , Ankita Bodkhe, Masrat Khan, Alfiya Firdaus , Neha Nitnware

Computer Science and Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

ABSTRACT

In this process, the goal is to identify and extract object features that have been commented on by the opinion holder and determine whether the opinion is positive, negative, or neutral. An objective sentence presents some factual information, while a subjective sentence expresses personal feelings, views, emotions, or beliefs. However, just knowing that sentences have a positive or negative opinion is not sufficient. This is an intermediate step that helps filter out sentences with no opinions and helps determine to an extent if sentiments about entities and their aspects are positive or negative.

Keywords : Sentiment analysis, Lexicon learning, WordNet, Appraisal theory.

I. INTRODUCTION

We'll be building a real-time chat application in which we are suggesting emojis and text based on the context of messages received.Using our application users can get emojis and text suggestions as replies to received messages while engaging in a chat.We are implementing the Pusher's pub/sub pattern to ensure the security of the system.We'll be using Angular JS technology for developing real time chat application.This chat application is basically used for chatting purpose and allow it's user to chat with their friend circles to join this chat system.Using sentiment analysis, we can suggest emojis to be used as replies to messages based on the context of the received message.Whereas using text analysis, we can suggest text to be used as replies to messages based on the context of the received message.

Chat Application has become a day-to-day utility for everyone. The reason for choosing this chat application is that it provides a good scope for beginners to implement a network based system. IC is a type of chatting application that provides text-transmission over the Internet. Chat Application operates in a similar way as that of a LAN Messenger over a Local Area Network. Messages are transmitted between two parties i.e. The sender and the receiver, it can also be between more than two parties (group chatting). The messages transmitted are bi-directional in nature. Messaging applications also use push technology to provide real-time transmission of messages as they are composed, character by character.

II. MATERIAL AND METHOD

TECHNOLOGY USED

AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

AngularJS is the frontend part of the MEAN stack, consisting of MYSQL database, Express.js web application server framework, Angular.js itself, and Node.js server runtime environment.

DATABASE

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).

SUPPORTING TECHNOLOGY

Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which uses the proprietary .sln solution file and project-specific project files. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

1) AFINN-BASED SENTIMENT ANALYSIS

AFINN is a list of words rated for valence with an integer between minus five (negative) and plus five (positive). Sentiment analysis is performed by cross-checking the string tokens (words, emojis) with the AFINN list and getting their respective scores. The comparative score is simply: sum of each token / number of tokens. So for example let's take the following:

I love cats, but I am allergic to them.

That string results in the following:

```
{
  score: 1,
  comparative: 0.1111111111111111,
  calculation: [ { allergic: -2 }, { love: 3 } ],
  tokens: [
    'i',
    'love',
    'cats',
    'but',
    'i',
    'am',
    'allergic',
    'to',
    'them'
  ],
  words: [
    'allergic',
    'love'
  ],
  positive: [
    'love'
  ],
  negative: [
    'allergic'
  ]
}
```

- Returned Objects

Score: Score calculated by adding the sentiment values of recognized words.

Comparative: Comparative score of the input string.

Calculation: An array of words that have a negative or positive valence with their respective AFINN score.

Token: All the tokens like words or emojis found in the input string.

Words: List of words from input string that were found in AFINN list.

Positive: List of positive words in input string that were found in AFINN list.

Negative: List of negative words in input string that were found in AFINN list.

In this case, love has a value of 3, allergic has a value of -2, and the remaining tokens are neutral with a value of 0. Because the string has 9 tokens the resulting comparative score looks like: (3 + -2) / 9 = 0.111111111

This approach leaves you with a mid-point of 0 and the upper and lower bounds are constrained to positive and negative 5 respectively (the same as each token! 😸). For example, let's imagine an incredibly "positive" string with 200 tokens and where each token has an AFINN score of 5. Our resulting comparative score would look like this:

(max positive score * number of tokens) / number of tokens
(5 * 200) / 200 = 5

2) Tokenization

Tokenization works by splitting the lines of input string, then removing the special characters, and finally splitting it using spaces. This is used to get list of words in the string.

III. RESULT AND DISCUSSION

Module 1: Login and Signup

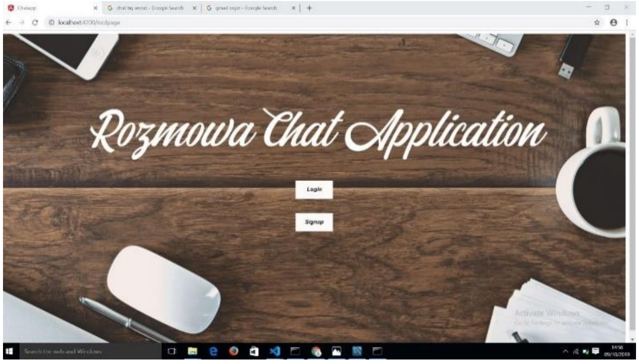
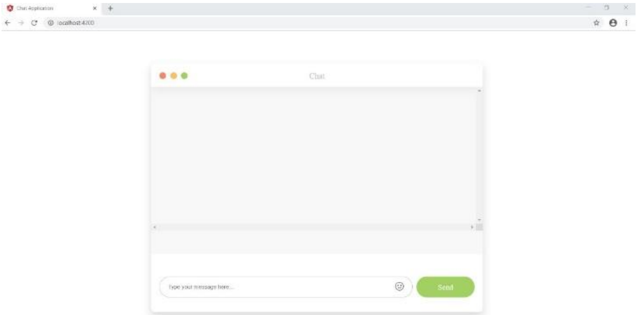


Fig (a): Home Page of proposed system

The home page shows all the basics of the task which the user wants to perform it can be login or signup for chatting.

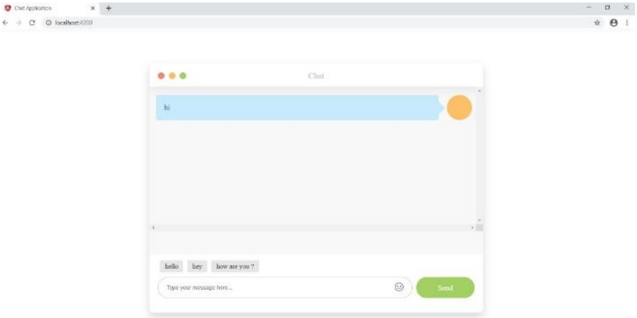
Module 2: Chat Application Module



Fig(b): Chat Page

The chat sessions are stored locally on the client computer. The user can easily review the messages that are exchanged by using the chat history. The main use case is: user A sees B online, A sends a message to B, B receives it.The secondary use case is: user A sees B offline, A sends a message to B. When B comes back online, B receives the message. The goal is to minimize latency. Speed matters. The messages should arrive in order. We cannot lose messages but receiving duplicates once in a while is fine. Just text data, no binary data

Module 3: Text Analysis Module



hi

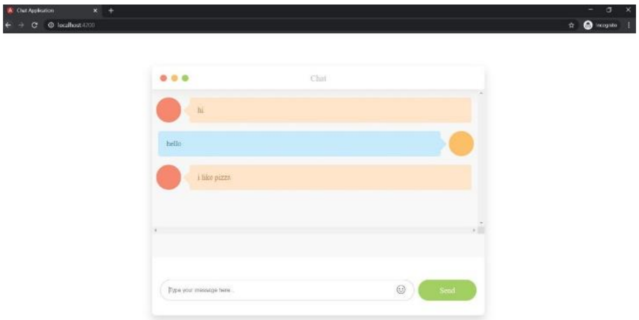
hello

hey

how are you ?

This module suggests the text while chatting.

Module 4: Sentiment Analysis Module



hi

hello

i like pizza

This module suggest the emojis based on the context of messages recived.

IV. CONCLUSIONS

Finally, to conclude, a prototype chat application which demonstrates how real time sentiment analysis can be used to detect user's mood by analyzing the chat messages has been developed.

In its current form, it is a prototype of a chat application having Real-Time Sentiment Analysis capabilities.

V. REFERENCES

[1]. A study of internet instant messaging and chat protocols published on 14 August 2006 by R.B.Jennings, E.M Nahum, D.P Olshefski, D Saha, Zon-yin Shae, Christopher J. Waters(http://ieeexplore.ieee.org/document/1668399/) https://en.wikipedia.org/wiki/Instant_messaging

[2]. Robert W. Sebesta: Programming the World Wide Web,8th Edition Pearson Education,2015

[3]. Behrouz A. Forouzan: Data Communications and Networking, 5th Edition, Tata McGraw-Hill,2013.

[4]. ity, Srinagar Garhwal.