IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

Smart Recruitment System

Siddhi Khanvilkar[1]*, Suparna Shetty[2], Disha Solanki[3], Sarika Davare [4]

[1,2,3,4]Dept. of Information Technology, Fr. Conceicao Rodrigues College of Engineering, Mumbai University, Mumbai, India

**Corresponding Author: siddhikhanvilkar1@gmail.com, Tel : +91 9969698054*

DOI: https://doi.org/10.26438/ijcse/v7i4.823828 | Available online at: www.ijcseonline.org

Accepted: 19/Apr/2019, Published: 30/Apr/2019

Abstract— nowadays a lot of organizations are in a constant lookout to simplify their hiring process so that they can scout the best talent in a minimum time frame. The proposed system tries to simplify the manual work by automating the entire hiring process. The system helps to clean, parse and classify the large amount of resumes, that the hiring managers' receive on a daily basis, using SVM (support vector machine algorithm in python).The system would cover some repetitive manual procedures like the aptitude test using JavaScript and the audio HR interview using natural language processing and sentiment analysis. This would ensure that the HR managers would not have to ask the same questions repeatedly thus preventing them from losing good candidates due to lack of interest towards the end of the interview. The system also provides detailed analysis in the form of a bar graph which gives a score count of the analysed tone parameters on the basis of the audio interview provided by the candidate.

***Keywords*—**Natural Language Processing , Support Vector Machine, Tone Analysis, Resume, Classification, Sentiment Analysis.

I. INTRODUCTION

Employing right candidate has always been an important but tedious task within any organization. The hiring managers need to manually sort through hundreds and thousands of resumes on a day to day basis and some good resumes may unintentionally get missed out while trying to sort through so much data. Also a large amount of resources are used to conduct aptitude tests for different candidates in different locations. Apart from this even HR interview rounds tend to be repetitive process as the same set of questions are asked over and over again. This results in lack of interest of the interviewer after taking multiple interviews thus missing out on the good candidates. Hence the proposed system would help automate the entire process.

The proposed paper is organized as follows, Section I contains an introduction of the proposed system which aims at automating the recruitment process and increasing the probability of hiring the right candidate, Section II contains literature review, Section III explains the workflow of the proposed system, Section IV explains resume parsing module which classifies the resumes based on keywords and job categories to which a candidate is best suited for, Section V explains audio and speech analysis of candidate using sentiment analysis, section VI and Section VII summarizes results and future scope.

II. LITERATURE REVIEW

Our research work for the existing system and software solutions for resume parsing module includes analysis various research papers. One paper on automated filtering of resumes [1],[2], queried the resumes using job specifications and then used collaborative filtering method to rank the relevant resumes. However, the system could not track the profiles of the candidates using social media sites. It could not compare applicant's resume with resumes of successful employees. Another paper used Term Document Matrix for extracting relevant words from the resumes.

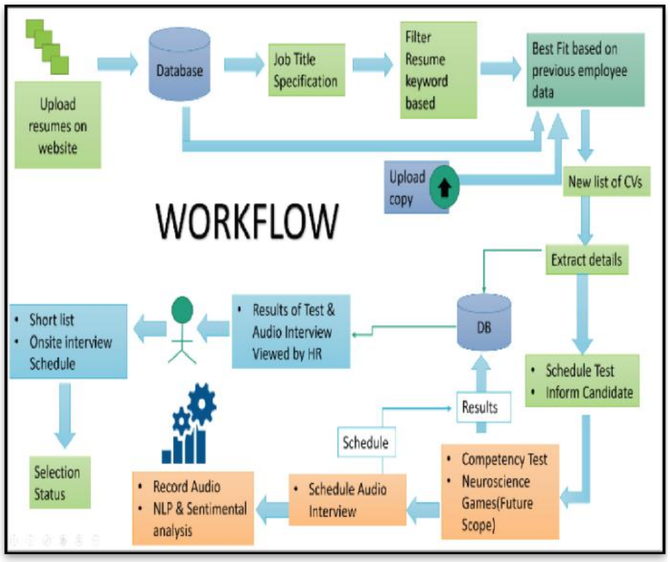
K-Means Clustering has been used in order to cluster the resumes having relevant word counts as features.

Each word Count C(i) was then multiplied with the corresponding weight W(i) and all such products for a resume were summed to get the Cluster Based Ranking (CBR). However this approach could not work on large dataset [3].

In order to understand the process and existing system, to perform analysis on text and audio we referred to three papers one of which performed sentiment analysis on the data that it received from twitter. This data was divided on the basis of positive text and negative text. The only problem was that it could not guess when certain emotions like sarcasm used positive text to represent negative emotions [4]. The next paper performed analysis on YouTube data by first converting video to audio and then converting audio to text and finally performing analysis on the text [5],[6]. We also referred to a paper that performed analysis on online papers in order to analyze which papers are better when compared to the others [7].

III. WORKFLOW

Initially the resumes are stored in the database in pdf format. For analysis we convert the pdf to text using pdf miner. Cleaning and analysis is performed on this text using pandas and NLTK word tokenizer library. Post which we extract details such as most frequently used words. We also extract frequently used adjectives which may create a good impression on a recruiters mind using the NLTK word tokenizer along with the POS tagging. Once the resume is selected we inform the candidate for further rounds. The candidate has to give an aptitude test, based on the results audio interview is scheduled. In the audio interview round based on the questions, audio is recorded of candidate's answer to those questions and simultaneously speech to text conversion takes place. Once these audios are uploaded, they can be viewed on the HR portal. Sentimental Analysis of the text is done using Tone analyzer. The result of the tone and sentiment analysis is displayed on the HR portal in a graphical format using bar chart. Based on the results, onsite interview may be scheduled and selection of the candidate can be made.



Upload resumes on website
Database
Job Title Specification
Filter Resume keyword based
Best Fit based on previous employee data
Upload copy
New list of CVs
WORKFLOW
Extract details
DB
Results of Test & Audio Interview Viewed by HR
Short list
Onsite interview Schedule
Schedule Test
Inform Candidate
Results
Schedule
Selection Status
Record Audio
NLP & Sentimental analysis
Schedule Audio Interview
Competency Test
Neuroscience Games(Future Scope)

Figure 1: Process Flow

IV. RESUME PARSING

The resume parsing model goes through the following stages:-

A. Conversion from PDF to Text

It is extremely complicated to perform processing functions on data when it is in a PDF format as PDF is an image representation without any logical structure. Hence in order to simplify the process we convert PDF to text. In order to do so we use PDF miner which is a library written in Python used to convert pdf documents to HTML format.

It uses a method called lazy parsing which means that it will only parse the important parts in a particular document. In order to use this, you need to make use of the various classes present in the library.

Two main classes are PDF Parser and PDF Document. The PDFParser gets the resume details from the resumes and PDFDocument stores this information in the form of objects. The PDFParser can also request for these objects whenever required. The PDFInterpreter performs various processing functions on the contents of the document. The PDFDevice then interprets the resume contents as per our requirements. Finally the text file is displayed.

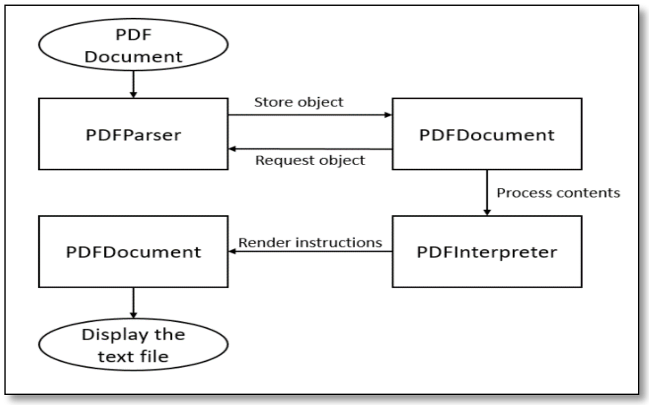


Figure 2: Procedure for conversion from PDF to Text

B. Cleaning and Preprocessing of Data

Our main aim for this part is to get rid of the redundant data and extract text which can be further used for analysis.
Removing punctuation and stop words. For our analysis we need to get rid of punctuations and stop words as they are of no use to us in our analysis example, Email, phone Nos in 'xxxx' format: Resume consists of personal details like phone numbers, emails, etc but due to some privacy reasons they are censored. This censored data may add noise to the dataset. So they are of no use to us. Since they follow a standard format we deal with it using regular expressions. Resume may include many UTF-8 format specific codes which aren't interpretable in ASCII. Also they are not much use to us as it adds noise.

So conversion to ASCII is done along with it UTF-8 format codes are removed. There may be some records wherein resumes may not be present. So for the analysis those records were removed.

C. Analysis

1. Class-wise Distribution

Based on the Category of the resume we have divided the classes in our database.

Some of the domain specific classes are HR, Finance, Engineering, Advocate, etc.

2. Frequency of words in the resumes

To identify the most frequently appearing words, the proposed system uses the NLTK-word tokenizer function.

For visual representation of these words our system uses word cloud. Some of the most frequent words are program, work -experience, project, etc appear many times, which can be useful in further analysis.

3. Frequency of adjectives

To create a good impression of his/her resume in the recruiter's mind, use of good adjectives is really common, rather important and the frequency of adjectives used gives us more insight into our dataset. The system makes use of the NLTK word tokenizer along with the POS tagging to get the parts of speech. Our output shows that words like professional, responsible, academic, technical, social which are quiet prominent in the word cloud, could have been mainly used in describing oneself. Our output shows that words like Annual, various, international, medical, senior, legal, public etc could have been used to explain about the past designations/jobs/income etc.

D. SVM Classifier:

In the proposed system we have made use of supervised learning model - Support vector machine (SVM) for classification of resumes into job categories, to which the candidate's profile is best appropriate for.

Implementation of SVM Classifier for resume classification:

1. Prepare and pre-process the resume.csv dataset

SVM classifier cannot directly process the text documents in original format. It expects a numerical feature vector of fixed size. Also, in this step we remove any punctuations, duplicate columns and convert the text into a more manageable representation.

2. Splitting the dataset into training and test data

In the proposed system, we are training the SVM model with a set of preselected resumes and categories of the job profile. We then use testing set of resumes on our model to predict their class.

3. Extracting Feature Vector.

SVM classifies data into classes based on extracted feature vector. We are using bag of words model which detects how many times the relevant words appear i.e their frequency count.

CountVectorizer-

For a given collection of text documents, CountVectorizer tokenises it into a vocabulary of known words and finds out their word count.

TF-IDF Vectorizer.-

Term frequency finds how often(frequency)of given words appears across documents whereas Inverse-Document Frequency scales down words that appear a lot across document for eg .the.

4. Building SVM Classifier.

We then build our SVM classifier by specifying labels. In the proposed system, SVM uses word and frequency score count of relevant words to classify them into a particular job category. For eg. For 'Advocate' job category, 'law' is a relevant word.

SVM Classification model was able to efficiently predict the job categories of test data and gave an accuracy of 70%.The classifier was able to work on non-linear and multivariate numeric data due to appropriate kernel selection.

Evaluating SVM Classifier:

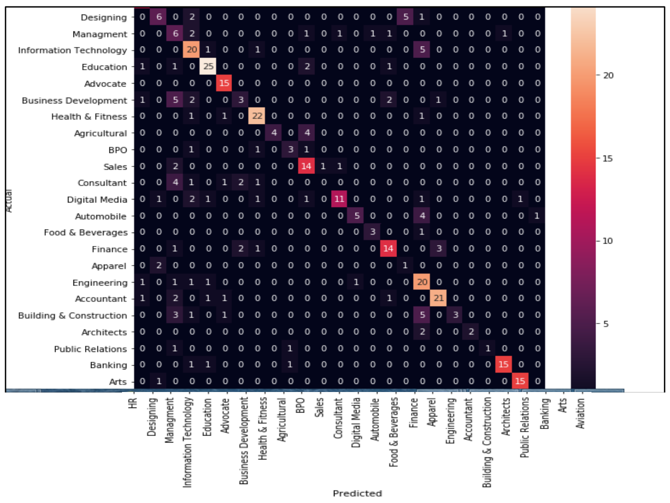
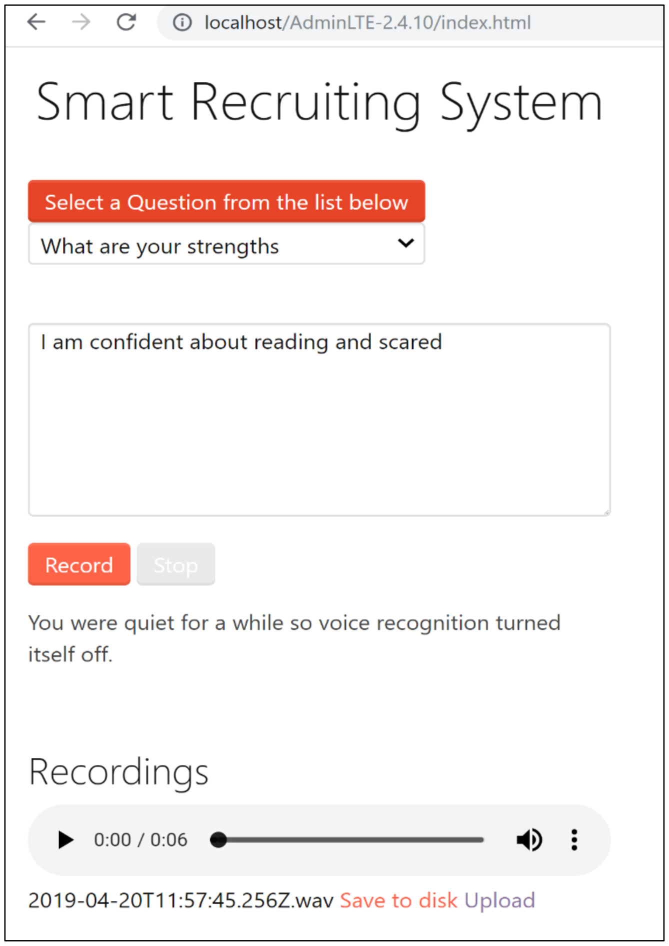


Figure 3: Mapping of actual and predicted values

As seen in the above diagram, most of the actual and predicted values of job categories lie on the diagonal which represents true positive values.

V. AUDIO INTERVIEW

The candidate has to first select a question from the drop down list. After clicking on the record button audio recording will start. As the recording is in process, simultaneously the speech gets converted to text. On pressing the stop button the recording terminates and audio gets displayed. When the upload button is clicked the recording gets uploaded to the database. On the HR PORTAL the audios' can be viewed along with the results of the text analysis. Results of the text analysis are shown in a graphical format.


localhost/AdminLTE-2.4.10/index.html

Smart Recruiting System

Select a Question from the list below

What are your strengths

I am confident about reading and scared

Record Stop

You were quiet for a while so voice recognition turned itself off.

Recordings

0:00 / 0:06

2019-04-20T11:57:45.256Z.wav Save to disk Upload

Figure 4: Audio Recording and Speech to Text Conversion

A. Speech to text conversion

In order to perform analysis on the interview given by the candidate, we need it in textual format. In order to do so, we need to convert the interview which is in audio format(.wav) into textual format. For this we make use of web speech API using JavaScript. When the candidate speaks into the microphone the speech gets converted into text using Speech Recognition and this text is displayed on a text box. When candidate clicks on the stop button, the textual conversion of audio data is sent to the database and on clicking the upload button the page gets refreshed for the next question to be answered.

1. In depth working

This API works only on Chrome and Firefox and hence the first part of our code displays an error message if any other browser is used. Next we specify a recognition variable which allows us to use all the API methods. The "recognition.continuous()" method is set to true as candidates tend to take pauses between words to think and this will ensure that the recognition does not stop abruptly. The recognition will stop automatically, when the candidate pauses for longer than 15 seconds. In order to display the speech in the text box we make use of recognition.onresult(). When we tried to run this application on a mobile all words were repeated twice. In order to prevent this we made some changes in the code by introducing a new variable "mobilerepeatbug". In order to start the recognition we used "recognition.start()" and in order to stop the recognition we use "recognition.stop()".

B. Tone Analysis of Audio Data:

The proposed system makes use of IBM Watson Tone analyser API which performs natural language processing and tone analysis by sensing emotional and language tones using linguistic analysis on textual conversion of spoken speech. In the proposed system we have used tone analysis API on the answers spoken in audio interview to understand how confident a candidate is. It Analyses tone (sentiment) into attributes like: fear, joy, anger, sadness, analytical, tentative, confident.

Since the tone analyser API works only on textual data we have stored speech to text converted data of spoken answers in our database.

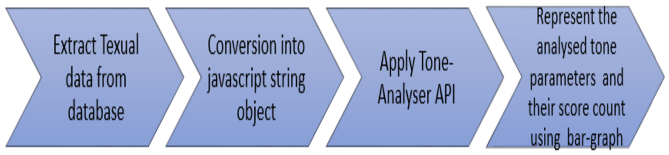
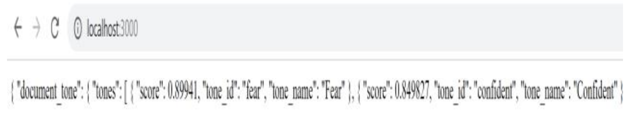

Extract Texual data from database
Conversion into javascript string object
Apply Tone-Analyser API
Represent the analysed tone parameters and their score count using bar-graph

Figure 5: Step by step execution of tone Analysis

Brief working of Tone-Analyser API:

1. We first extracted the speech to text converted data
2. (textual data) of answers spoken in audio interview from database.
3. We then used JSON.parse() to convert it into JavaScript object (string format).
4. We then iterated through each sub item in JavaScript string object and passed this object to tone analyser API.
5. The tone analyser then analyses the string object into parameters: fear, joy, anger, sadness, analytical, tentative, confident and along with that score count of each analysed tone parameter.
6. It then converts the analysed tone parameters into JSON object using JSON.Stringify() and views the JSON analysis of speech.



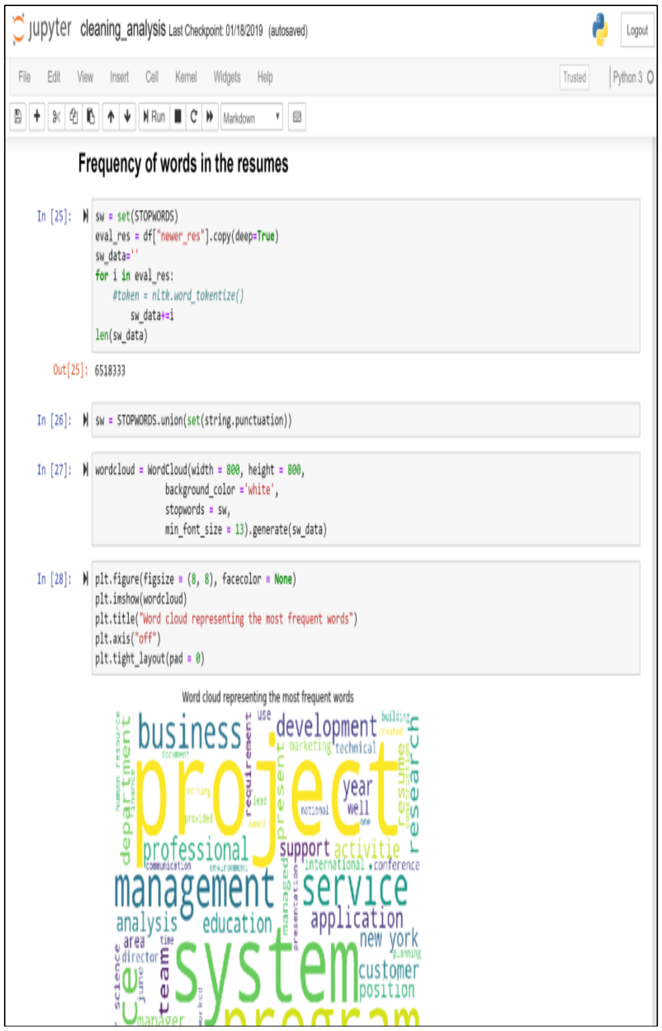
{ "document_tone": { "tones": [{ "score": 0.89941, "tone_id": "fear", "tone_name": "Fear" }, { "score": 0.849827, "tone_id": "confident", "tone_name": "Confident" }

Figure 6: Result in Json Format

We later used Chart.js to represent the tone Analysis result using bar-graph for better understanding and representation.

VI. Results and Discussion

The proposed system therefore helps to improve and speed up the process of recruitment. The system can successfully parse large amount of resumes with proper classification on the basis of the role that the employees applied for, and has an accuracy of 70%.

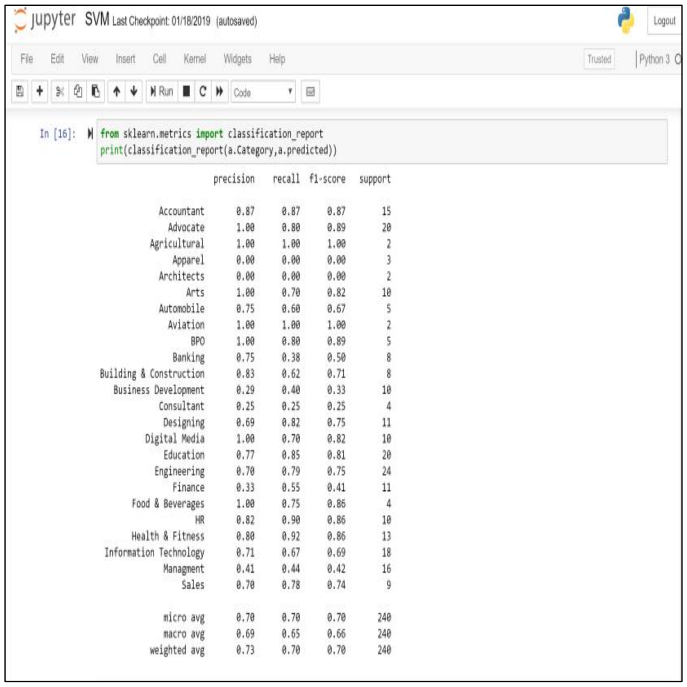


jupyter cleaning_analysis Last Checkpoint: 01/18/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown

Frequency of words in the resumes

```
In [25]:  sw = set(STOPWORDS)
          eval_res = df["newer_res"].copy(deep=True)
          sw_data=''
          for i in eval_res:
            #token = nltk.word_tokentize()
                sw_data+=i
          len(sw_data)

Out[25]:  6518333

In [26]:  sw = STOPWORDS.union(set(string.punctuation))

In [27]:  wordcloud = WordCloud(width = 800, height = 800,
                          background_color ='white',
                          stopwords = sw,
                          min_font_size = 13).generate(sw_data)

In [28]:  plt.figure(figsize = (8, 8), facecolor = None)
          plt.imshow(wordcloud)
          plt.title("Word cloud representing the most frequent words")
          plt.axis("off")
          plt.tight_layout(pad = 0)
```

Figure 7: Frequency of words represented by word cloud



jupyter SVM Last Checkpoint: 01/18/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [16]:  from sklearn.metrics import classification_report
          print(classification_report(a.Category,a.predicted))

                          precision    recall  f1-score   support

              Accountant       0.87      0.87      0.87        15
                Advocate       1.00      0.80      0.89        20
            Agricultural       1.00      1.00      1.00         2
                 Apparel       0.00      0.00      0.00         3
              Architects       0.00      0.00      0.00         2
                    Arts       1.00      0.70      0.82        10
              Automobile       0.75      0.60      0.67         5
                Aviation       1.00      1.00      1.00         2
                     BPO       1.00      0.80      0.89         5
                 Banking       0.75      0.38      0.50         8
 Building & Construction       0.83      0.62      0.71         8
    Business Development       0.29      0.40      0.33        10
              Consultant       0.25      0.25      0.25         4
               Designing       0.69      0.82      0.75        11
           Digital Media       1.00      0.70      0.82        10
               Education       0.77      0.85      0.81        20
             Engineering       0.70      0.79      0.75        24
                 Finance       0.33      0.55      0.41        11
        Food & Beverages       1.00      0.75      0.86         4
                      HR       0.82      0.90      0.86        10
        Health & Fitness       0.80      0.92      0.86        13
  Information Technology       0.71      0.67      0.69        18
               Managment       0.41      0.44      0.42        16
                   Sales       0.70      0.78      0.74         9

               micro avg       0.70      0.70      0.70       240
               macro avg       0.69      0.65      0.66       240
            weighted avg       0.73      0.70      0.70       240
```

Figure 8: Evaluating SVM Accuracy

The proposed system also features audio interview module which uses natural language processing and sentiment analysis to detect emotional and linguistic tones in the candidate's speech. The analysed tone parameters with their score count are represented by a bar–graph for better understanding of the recruiter.

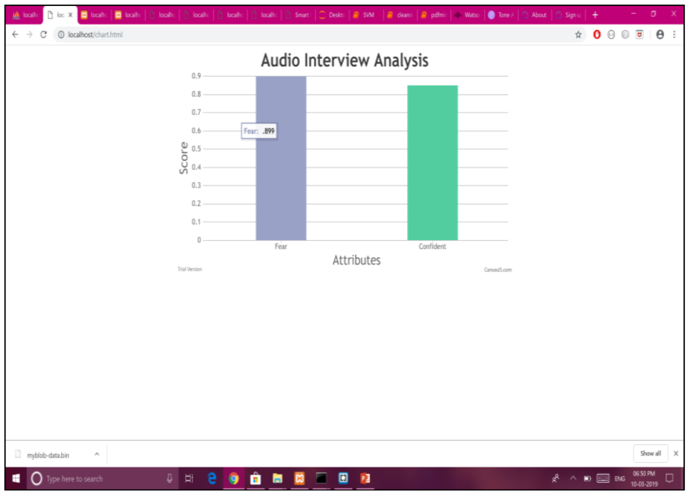


Figure 9: Chart representation of tone parameters

VII. Conclusion and Future Scope

The scope of the proposed system could be increased in future by including advanced competency tests which will detect candidate's aptitude and cognitive skills along with emotional intelligence with the help of fun games [8]. These neuroscience games can help detect problem solving, critical thinking and time management abilities. They can also help recruiters to assess how willing a candidate is to take risks. We could also use face recognition technique to check the authenticity of candidate. We can also have a Chat Bot which could be used to solve all queries of the candidate. We can also have a calendar feature to schedule and book dates for different kinds of interviews.

ACKNOWLEDGMENT

We would like to thank our project guide, Mrs.Sarika Davare for her constant motivation and guidance throughout the project. We would also like to thank Fr. Conceicao Rodrigues College of Engineering for this wonderful opportunity.

REFERENCES

[1] Prarthita Das and Amala Deshpande," *Automated Filtering of Relevant Resumes*" International Journal of Computer Applications Volume 154 – No.6, November 2016

[2] Sunil Kumar Kopparapu, "*Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search*", published in IEEE 2010.

[3] Mayuri Verma "*Cluster based Ranking Index for Enhancing Recruitment Process using Text Mining and Machine Learning*" International Journal of Computer Applications (0975 – 8887) Volume 157 – No 9, January 2017

[4] Apoorv Agarwal, Boyi Xie Ilia Vovsha, Owen Rambow Rebecca Passonneau, "*Sentiment Analysis of Twitter Data*" IEEE ICASSP journal

[5] Lakshmish Kaushik, Abhijeet Sangwan, John H. L. Hansen," *SENTIMENT EXTRACTION FROM NATURAL AUDIO STREAMS* ", CP-ICASSP13

[6] Mishne and Glance, "*Predicting movie sales fromblogger sentiment,*" in AAAI 2006 Spring Symposiumon Computational Approaches to Analyzing Weblogs, 2006.

[7] *Online paper reviews using sentiment analysis*, (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 6, No. 9

[8] White paper-pymetrics: Using Neuroscience and Data Science to Revolutionize talent Management.

[9] Ketan Sarvakar and Urvashi K Kuchara," *Sentiment Analysis of movie reviews: A new feature-based sentiment classification"* International Journal of Scientific Research in Computer Sciences and Engineering Vol.6, Issue.3, pp. 8-12 , June (2018).

[10] Amit Palve, Rohini D.Sonawane ,Amol D. Potgantwar," *Sentiment Analysis of Twitter Streaming Data for Recommendation using Apache Spark*" International Journal of Scientific Research in Network Security and Communication Volume-5, Issue-3, June 2017

Authors Profile

Ms. Siddhi Khanvilkar is pursuing Bachelor of Engineering in Information Technology from Fr.Conceicao Rodrigues College of Engineering, University of Mumbai.

Ms. Suparna Shetty is pursuing Bachelor of Engineering in Information Technology from Fr.Conceicao Rodrigues College of Engineering, University of Mumbai.

Ms. Disha Solanki is pursuing Bachelor of Engineering in Information Technology from Fr.Conceicao Rodrigues College of Engineering, University of Mumbai.

Mrs. Sarika Davare is the assistant professor of the Department of Information Technology of Fr.Conceicao Rodrigues College of Engineering, University of Mumbai.She has completed her M.Tech in Information Technology. Her areas of interest are Database Systems, Data Mining and Machine learning.