Resume Scanning and Emotion Recognition System based on Machine Learning Algorithms

[1]Vishruth R G, [2]Sunitha R, [3]Varuna K S, [4]Varshini N, [5]Prasad B Honnavalli
[1,3,4] Student, Department of Computer Science Engineering
[2] Assistant professor, Department of Computer Science Engineering
[5] Professor, Department of Computer Science Engineering
[1,2,3,4,5] PES University, Bangalore, Karnataka, India
vishruth1020@gmail.com, sunithar@pes.edu, varunaks18@gmail.com, varshini.varsh004@gmail.com, prasadhb@pes.edu

Abstract: In the current smart world, everything should be done faster, smarter, and accurate way. The various organization's recruitment processes will be done face to face in an arranged venue. But, during some pandemics like Covid-19 face to face recruitment process will be very difficult. In the proposed system, a smarter way of performing the recruitment processes anywhere around the world based on the company requirements is performed. The aim of this article deals with making the process of candidate recruitment easier for companies. The amount of manual work that goes into recruiting processes is reduced and the initial scanning process of candidates was performed. By eliminating the redundant candidates helps in retaining only the applicable ones. Achieve this through the help of resume scanning, initial aptitude testing of candidates, and an interview session where the candidate answers questions asked by the interviewer. With this model, all the time and manual labor that is wasted in eliminating the redundant candidates is accomplished. It chooses the one who is best applicable to a job by comparing it with the job description based on the resumes received. Our model is working accurately for some of the predefined parameters of the company in a recruitment process by providing more security and reliability.

Keywords: Resume Scanning, Neural Networks, Chatbot, Emotion recognition, Data Preprocessing.

1. Introduction

is the proposed system aids to design a website that helps to smoothen the process of applying for jobs. Subsequently, 200,000 applications are received for 30,000 jobs in a week at a rate of 18 applications per minute. When processing these manually is near impossible. Many a time the interviewers face trouble finding the compatibility of a candidate with the company. Existing systems are inefficient. Application Tracking System (ATS) is based on keywords, easy to manipulate, takes time to learn, any activity that happens is always public.

The amount of manual work required to filter out the irrelevant candidates with the help of these 3 modules. In this model, an interactive web platform is designed for the users. It is a simple, self-explanatory, and interactive platform that helps both candidates and organizations make the process of applying for a job and short listing of candidates for a job hassle-free and easy.

The resume scanner is a feature that helps us to compare job descriptions given by the organization and the resume submitted by the candidates. Through this comparison, found out which are the most relevant resumes and shortlist those candidates. Depending on their rank, the candidates can then proceed to round 2, the interview round. Here, a chatbot specifically designed to train and help the candidates to find platforms to help them practice for interviews according to the company of their choice and help them get in touch with the organization in case they have any further questions related to a job. Our emotion detection model helps us to evaluate the candidate's confidence levels and accordingly give them a score. This model helps us to determine how apt a candidate is for a job. The data collected are stored in the form of documentation for future references and Python is used for the scripting of the server.

The main technologies used are Convolutional Neural Network [13] (ConvNet/CNN) with the Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. A deep network [14] is when there are two or more such layers of neuron between the input and the output. Neural networks have been used for the chatbot as well as the video interview modules of the article. The companies who need to shortlist few among the many applications they receive. Here, Nltk stands for natural language toolkit. This is a popular python framework which allows us to deal with processing and training words. This is used in designing resume screening and the chatbot module. Nltk library is very useful in finding out the context of words and the relevance of the said words. It is useful to identify stop words and bring out the concept of term frequency and inverse document frequency. Both the concepts are very essential to the understanding of word relevance in documents and across documents.

Flask is a python web framework that works on RESTful APIs. Flask is very popularly used for designing servers. Flask is used to implement the server-side code for the paper. Flask is very useful in routing and ensuring that there is no mistake in understanding the process flow of a website. It is essential to interact with the MySQL database and hence route the users to their respective dashboards and display the details related to their profile. The proposed work can be used by job seekers and companies alike to help with their hiring process and ensure that the process is not time intensive. The majority groups of users who face this problem include active and passive people. The active people who are currently unemployed and looking for opportunities are immediately available and the passive people who are already employed but are interested in moving.

2. Literature Survey

The proposed model mainly revolves around the amount of data that needs to be processed before a decision about giving a candidate the job can be made. Around 2,00,000 applications for 30,000 need to be processed in a week at a rate of 18 applications per minute. Often due to the huge amount of data, interviewers face trouble finding the compatibility of candidates with the organization. As a result of the huge amount of data, they are unable to get the cream of the crop. To make this easier designed an application to cater to this need.

V. Yadav, U. Gewali, S. Khatri, S. R. Rauniyar and A. Shakya[1], This paper suggests and discusses the need for colleges to have an online job board program and its usefulness in bridging the gap between college students and employment opportunity. Job platforms have historically been used in HR management to find applicants and in recruiting. This research is focused on a career portal constructed for one of Nepal's leading engineering campuses, the Pulchowk Campus. E. Bagarukayo and E. Mwesigwa [2], To achieve this goal, they built a mobile app that aggregates existing web portal job posts to improve usability, timeliness, and productivity. The mobile application (app) was built using agile methodologies especially Scrum and eXtreme Programming. Results showed that the software not only saves money for employers but also offers a reliable and realistic way of connecting job search. A. Mohamed, W. Bagawathinathan, U. Iqbal, S. Shamrath and A. Jayakody [3], this work aims to familiarize ourselves with the actual results they have achieved on a new recommendation system called Smart Applicant Ranker which is a candidate recommendation tool designed to supervise recruiters as they insert their job requirements into the system. This framework is designed using Ontology to compare the resume models with the work specifications to suit the best possible.

Bing Liu, Gokhan Tur, Dilak Hakkani-Tur, Pararth Shah, Larry Heck.[4] This paper was developed at Carnegie Mellon University in partnership with Google Research and Samsung Research. The authors describe a novel learning method for training task-oriented dialogue systems through online user interactions. A task-oriented dialogue system aids a user to complete tasks by carrying out a back-and-forth communication between the system and the user. The architecture used in this work involves an LSTM (Long Short-Term Memory) neural network to encode dialogue with a series of turns. Depending on the user input, the neural network generates probabilities of the next optimal state. The state (response) with the highest probability is presented to the user. Training of the model is done on end-to-end dialogue samples. A significant contribution of this paper is the attempt to imitate learning with human teaching. In this manner, the system improves through communicating with the user and learning from the mistakes in the process. The dialogue training itself happens through reinforcement learning. This paper laid the basics of developing the chatbot used in this paper.

Florian Schroff, Dmitry Kalenichenko, James Philbin [5] developed at Google Research, this paper introduces a new system called FaceNet. This was devised to overcome the challenges of implementing face verification and recognition at scale. The work aims to unify the verification recognition and clustering of faces. Based on Euclidean embedding, their method user a deep convolutional network to achieve the goals. The method is to batch normalize the training set and pass it through the neural network. This is then passed through L2 normalization and the loss function used is the triplet loss function. The CNN itself was trained using the Stochastic Gradient Descent (SGD) and AdaGrad. These concepts were of very significant aid for us in developing the facial recognition and emotion recognition techniques used in the paper.

Sumit Maheshwari, Abhishek Sainani, P. Krishna Reddy [6] designed an approach for resume Screening and selection is the main highlight of this paper. It uses a hierarchical structure of the resume to screen them efficiently. The work done here makes use of three layers to structure the resume. Beginning from the most common features, the subsequent layers contain more and more specialized features. Through this layered architecture, the system extracts skill type and skill values (a key-value architecture) that is stored. For each layer, the Skill Type Features (STFS) and Skill Value Features (SVFS) are stored. That is, for each level (Common, intermediate, and Specialised), there exists a separate STFS-SVFS table. By making use of such values, it becomes much simpler to screen the resumes based on job descriptions. This paper showed us ways to implement our resume screening algorithm. Although this exact method of table entries was not used in our paper, it leads to the development of a “vectorization” of such common, intermediate, and specialized features in the resumes to screen them more accurately and efficiently.

Kun Yu, Gang Guan, Ming Zho [7], developed in Beijing, China. Like the work done by Sumit Maheshwari et al, this work also presents a hierarchical structural of storing values for the details in the resumes based on the specialty of the skills. The main aim of this work is the comparison in the use of the Hidden Markov Model (HMM) [5] and the Support Vector Machine (SVM) [6] in calculating probabilities. The model calculates the optimal probability by calculating the maximum likelihood estimations. Initially, the words of the resumes were vectorized using the TF-IDF vectorizer. This method was also used in our paper for the implementation of the resume screening code. From the use of the mentioned models to run the maximum likelihood estimations, it was determined that SVM performs better than the HMM. Due to processing issues, they have used a model like SVM, the nearest neighbours, to screen the resumes which best fit the given job description.

Octavio Arriaga, Paul Ploger, Matias Valdenegro [8] presented a unique way to classify genders using images. The architecture used is a convolutional neural network. The model is built such that, the input is many images. These images are normalized and fed into the convolutional network. At the end of each pa, the pooling is done using MaxPool and finally, a softmax function is used as activation.

Sungheetha, Akey, and Rajesh Sharma [15] designed aspect-level sentiment, a capsule model known as TransCap was presented. With the support of document-level data that is present in abundance, this work discusses the problems of data that are classified at the aspect level. Using simple routing algorithms such as dynamic routing and aspect routing, the information is transferred to the element level task from the document-level task. Comparing the performance achieved with other baselines shows that the technique works well. In reading the text and to recognize emotions such as sarcasm, the writers proposed that the technique could be further improved by introducing new technical developments.

The other major contribution of this paper is the emotional classification of the images. The model is trained on the images of faces against labeled emotions. This model serves as the tool to classify emotions for further test images. This paper has been used to implement the emotion recognition module of our paper. The major addition to this is to use video input instead of still images. In the further sections, the proposed model design, implementation, and results were discussed.

3. Proposed Work

Resume Scanning and Emotion Recognition System approach is designed for the various organizations. This system is divided into three modules and designed are: Resume scanning, Chatbot implementation, and Emotion Recognition.

The developed portal helps to smoothen the process of applying for jobs. A few constraints and assumptions for our model are given below.

a) The assumptions made involve that the resume documents of type pdf, doc, or docx are accepted. The job description uploaded must be of .txt format.
b) There is a constraint on the size of the document uploaded.
c) One of the risks involved is the possibility of words written in white font colour hence invisible to the naked eye but accepted by our program. This model aims to solve this by trying to check font colours. If any document is submitted as a screenshot of the resume, then it will not be accepted and parsed by our program.

3.1 Resume Scanning

In this module, performing document comparison and document ranking to filter out a completely unnecessary resume. First, perform a summarizing of both the documents. This summarizing happens based on the term frequency to find out the top few words of the document that helps to summarize the document. Once the term frequency array of each document is obtained, and then use inverse document frequency to find out which document is most relevant to the job description file. The top resumes that fit the job description are then selected and the candidate names are listed.

The overall process is depicted in the following steps:

1. In the resume screening window, the user needs to select the job description to be screened for. So, the job description file is compared with the resumes of the candidate that have applied for the job.
2. The next step is to remove the stop words from the files. This is done because stop words have a very high frequency and must not be chosen in place of the more relevant and important words.
3. The step following this is to summarize the resumes and the job descriptions.
4. Following this is the word vectorization step, where it uses the concept of tf-idf, i.e., term frequency-inverse document frequency. Term frequency helps to rank the words based on their importance within a document. This is done for all the resumes and the job descriptions. Once the ranking within documents is got, it needs to get the ranks of the words across documents. This is done through inverse document frequency.
5. After analyzing, how much a resume matches the job description using k nearest neighbors' algorithm. This helps to rank the resumes with the job description in the center of the map. With this, rank resumes based on how close they are to the job description.
6. The output obtained is a list of resumes ranked in order of relevance to the job description.

3.2 Chatbot

This module is implemented for the candidate's searching and navigation purposes. The module interacts with the user and when they ask a question then the chatbot replies with an answer. This is a neural network that learns as time goes on. The chatbot can also give links to websites that the candidates can refer to before preparing for their interview with the company. Some of the sample questions along with the answers are allocated to them. So, candidates can ask a question and it will be answered in the chatbot.

The overall process is depicted in flowchart 4 with the following steps.

1. The chatbot GUI window is displayed on the screen. This screen consists of a text area, a text display area, and an Exit button to quit the chatbot window
2. User inputs a query to the chatbot
3. If the user query is not the string "quit", go to step 4, else go to step 9
4. Accept user input query
5. Run each query into the trained deep neural network
6. The appropriate prediction from this model, print response using the intents JSON file
7. Return promptly to the user
8. If the exit button is not pressed, go to step 3, else go to step 9
9. End chatbot

Neural Network [12] Architecture: The architecture of the chatbot neural network is straightforward. The input to the network is the column-value of the training values. This column-value contains the TF-IDF vectorized set of strings. The input size is equal to the column length. There are two fully connected layers with 8 neurons each. These layers were architected such that, in the first layer, the network will learn to identify the various kinds of questions that come under a tag. The second layer learns to find the optimal response to the given pattern request. Finally, the SoftMax activation function is used s activation for the network. The entire architecture has been written in TensorFlow v. 1.16.

a) Training: Using this dataset, the neural network is trained for 1000 epochs. Through this phase, the network can identify the best answers to the user's queries. Once the training is completed, the model is saved to reduce training efforts repeatedly.
b) User interface: The user interface for the chatbot is simple. There is a window that opens whenever the hyperlink for the chatbot is clicked. This window contains a prompt space for the user to type in the query. This is taken as input to the network model and a suitable response from the model is extracted and printed onto the window. This goes on until the user either clicks on the "Exit" button or types "Quit".
c) Video Interview: The video interview module is to conduct a preliminary interview before inviting

applicants for an onsite interview. This module consists of an aptitude quiz and a behavioral question. The aptitude quiz consists of five randomly chosen aptitude questions with four options each. The behavioral question is given 150 seconds to answer. The applicant must record their answer to this question, this answer video will be analysed and a report for emotions is created.

3.3. Emotion Recognizer

In this module, conduct an online interview and then perform an emotion analysis on the video. The whole video is split into frames and then each frame is analyzed, and the emotion is classified into any one of the five basic emotions of happy, angry, sad, surprised, and neutral. All the emotions other than happy are given a score of 0 and happy is given a score of 1. In the end, the score is calculated and given. The emotion recognizer helps assign a score to the candidate depending on their performance. The emotions are logged in according to the candidate and saved. The emotions are logged in according to the candidate and saved, it is then passed on to the organization in case they require it. The overall process is depicted in flowchart 3 with the following steps.

1. Click on the Video Analysis module.
2. A drop box with the saved interview videos is displayed.
3. Select a video and perform analysis.
4. Frame by frame analysis of the video - Detect face in the frame.
5. Use a neural network to classify the emotions.
6. Store the output in the database.
7. End of the video analysis process.

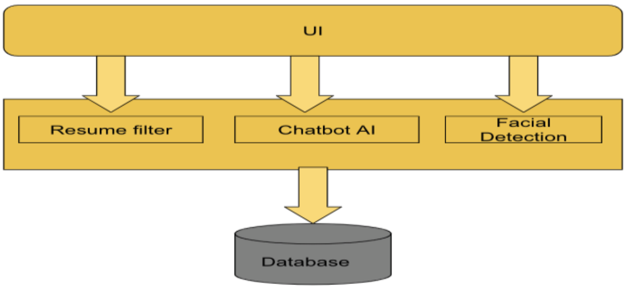

Figure 1 System architecture

This proposed system makes use of client-server architecture as shown in figure 1. The server is run with the Flask module in Python. This server is used to carry out all the back-end scripts for running various aspects of the paper. The four main functions of the server are the following:

a) Running the server python script to filter resumes based on the job description text
b) Running the python script for the training of the chatbot to be able to reply coherently to the questions asked by the user
c) Running python script to launch webcam for the interview module of the application
d) Reading from and writing to the database

The client side has the user interface to use all the modules in this article. Further, the UI allows login for the individual customer.

4. Results and Discussion

The resume screening process uses a company-provided job description to identify the most relevant resumes of candidates for the job. When the company provides a document containing the job description they want to hire for, this description is converted to metric form by calculating the value of the words in the description. Thus, the common words such as "a", "an", "the", etc can be eliminated. Once this is done, each of the words of the document is given a value corresponding to the rarity of the word.

First, the document is summarized using the summarize module in gensim (TextRank algorithm) then the TF-IDF vectorizer is used. There is no standardized template, the resumes can be uploaded in PDF, DOC or docx format. It will extract the text from the resumes using extractText for PDFs and textract.process for doc and docx. The database contains a set of questions that are randomly selected and presented for every candidate. The questionnaire prepared for the candidate will be displayed for each candidate and shuffling is done randomly. On the company's side, it is possible to select the top 3 candidate whose resumes are perfect for a particular job description, and then they are invited to attempt the quiz and facial analysis to find out how compatible they are for the company. The results of the quiz and facial analysis are uploaded to the database.

The database includes 2 different job descriptions to test and ensure that the resume scanning works for any type of job description and is applicable during a real-time scenario. This helped us to simulate a real-time use case scenario of our project in terms of resume screening especially. The more commonly occurring gets a lower value and vice versa. This is achieved by using the tf-idf vectorization. Tf stands for term frequency. Term frequency is the number of times a particular word occurs in a single document.

$$tf(t,d) = \text{count of } t \text{ in } d \,/\, \text{number of words in } d$$

Here d is the document and t is the term.

Idf stands for inverse document frequency. Document frequency means the number of times a term appears in the entire corpus of documents.

$$df(t) = \text{occurrence of } t \text{ in documents}$$

To normalize this function, the inverse document frequency is used to smooth it out with a logarithmic function. This gives us

$$idf(t) = \log(N/(df + 1))$$

N is the total number of documents in the corpus.

Once the vectorization is complete, a nearest neighbor algorithm is performed to identify the closest match the job description and the content of the resume. The 5 closest matches will be displayed on the webpage. When a user clicks on the dashboard button, they can navigate to the common dashboard page if they have not logged in. Once the candidate logs in to the webpage as displayed, the page shows the candidate's profile information on the right and the details of the jobs the candidate has applied for on the left. When a candidate scrolls down, he finds the list of all available jobs and the organization that are offering the jobs. Next to each job is an apply button that the candidate can use to apply for the job. Once the candidate logs on the webpage as displayed in Figure 2 is seen.

The web page shows the candidate's profile information on the right and the details of the jobs the candidate has applied for on the left. Upon clicking the apply button again for a job that he has already applied to the alert bar hence preventing the client from applying for it again. The window of the chatbot that is visible when the client clicks on the chatbot link on the top right corner of the candidate dashboard.

There is an input area at the bottom of this window where the candidate can type in the question and on clicking the enter button the chatbot answers the asked question. There is an exit button at the bottom that can be used to close the chatbot windows when an organization logs in, the organization details are displayed on the right whereas on the left the organization has the option to upload a job or remove a job.

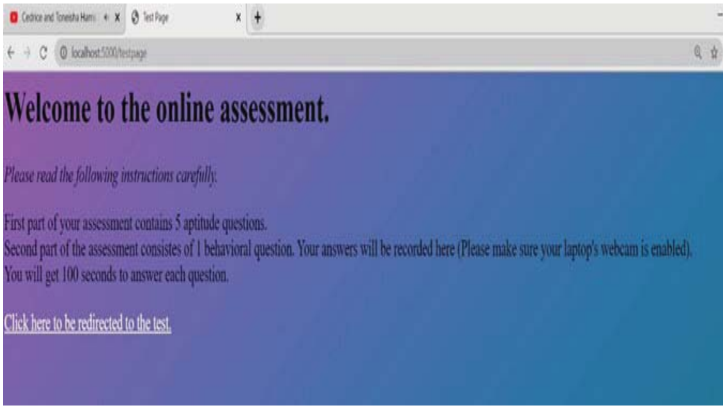

Figure 2 Online assessment

When a user clicks on the upload job button, they are shown the above fields. The organization name and email ID are fixed labels whereas the position, deadline, and job description fields are meant to be filled in by the organization. This ensures no company can post a job that does not belong to their company. When a user clicks on the remove job button, they are given the list of the open jobs and an input field where the name of the job description file needs to be filled in and then upon clicking the delete button, the job is deleted from the database. When an admin log in they are shown the dashboard hey can see the list of candidates that have registered with the website. Along with that, there is a list of organizations and the jobs that are offered by organizations. The admin has an option to register new companies with the website. He can also screen the resumes that have been submitted for a job.

The registration and login module allows users, both candidates as well as company executives, to login to their respective dashboards. These dashboards offer a bird's eye view of all the services offered to the users. It also allows easy navigation between sites.

The chatbot AI is a component provided for the candidates to help them prepare for interviews and help them practice with sample interview questions. To achieve this, a deep network was designed to train a model to give out a response depending on the input query given by the user. The dataset used was an intents json file. This file contains tags, sample requests and the standard responses to the hardcoded requests. Initial responses are used to train the model the score from these two parts is stored and candidates are ranked based on this as shown in figure 3.

For the emotion analysis, here the fer2013 dataset is considered with mini_XCEPTION model and Keras library and trained for 10000 epochs. As a result, 66% accuracy of prediction efficiency is accomplished. For this, used OpenCV to read the video frame by frame and analyze the emotions. First, convert the image to grayscale and use Haar Cascade to detect a face in the frame. Then apply emotion analysis on this image, based on the most probable emotion returned, score every frame.

Currently, it can recognize five emotions: Happy, Sad, Surprise, Anger, and Neutral. Higher scores are associated with positive emotions like Happy and surprise while negative emotions like anger and sadness are given lower scores. The average of this analysis is stored in the database along with the aptitude score.

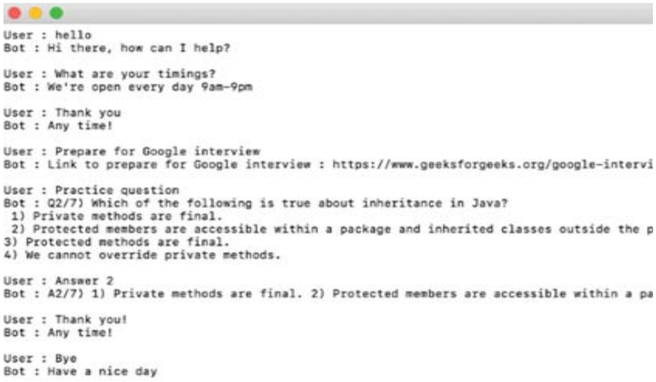

Figure 3. Chatbot

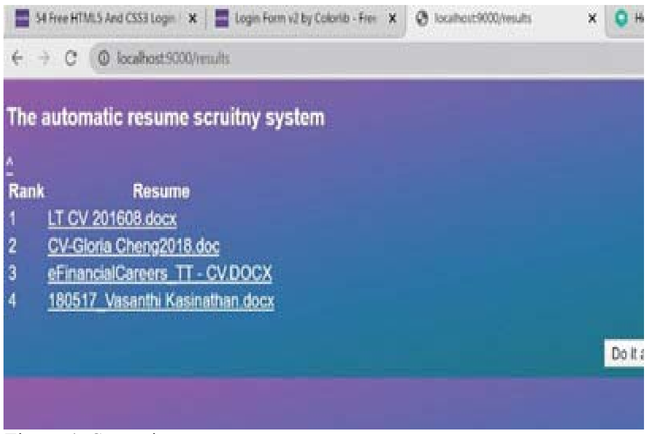

Figure 4. Screening

On performing application screening, the results as shown in figure 2 to 5 are obtained. It shows the list of applicant resumes that are suitable for the job according to the job description. There is an option to do it again in case an additional resume has been added or if the job description has been modified. Figure 4 shows the opening page of the test module. It shows the rules for the test. The link to the test page is also included in the end. Figure 5 shows the test page for aptitude questions. It contains five questions with four options each. A submit button that records your responses and takes you to the next part of the test module as shown in figure 5.

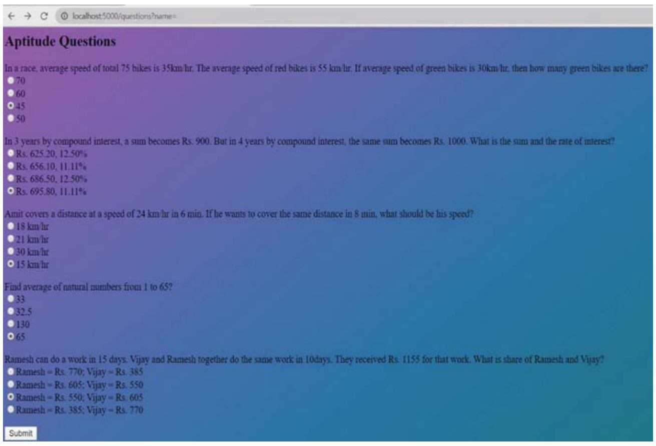

Figure 5. Questionnaire

The second part of the test module has a behavioral question. Here you will be given 150 seconds to answer the question. A link is given to record your answer.

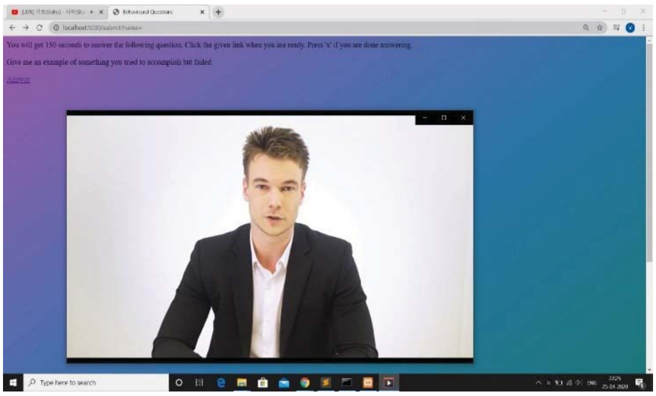

Figure 6 Video Interview

Figure 6 shows the answer recording for the behavioral question. It shows the webcam recording your answer. Once your answer for the behavioral question is recorded i.e. the test module is completed. It is the last page in the test module.

Two important features of our proposed smart recruitment model are Security and Reliability.

4.1 Security

The website is secure in terms that users can only access their own dashboards and features, they cannot access other user's dashboards. For example a candidate cannot create a job on behalf of an organization. An organization admin needs to login to create a job. Similarly, an organization can create jobs in their company's name. The passwords are also encrypted and stored so anyone with access to the database cannot log in as any other user. When an organization is created by the admin then a password is randomly generated and sent to the organization's registered email id.

4.2 Reliability

The website is reliable and other users cannot log in as any other user. It is hence reliable and behaves in the way it is supposed to. There is a continuous session variable that monitors the email ID and keeps track if the user is a candidate, admin, or organization user.

5. CONCLUSIONS

Smart recruitment portal is a very interactive and useful interface for the recruiters and user friendly for the candidates. There is very much required for the sentimental analysis and facial recognition mechanisms in the current trend of placements. Since this work helped in understanding the requirements of any candidate would need to be fulfilled when preparing for an interview. So, it is very illuminating and an interesting experience to explore many different options that are present as a solution for certain recruitment issues. The aim is to obtain more benefit on both candidates and organizations by making the portal self-explanatory and easy to navigate. This portal was meant to reduce the amount of time required to process resumes and help candidates effectively prepare for interviews and questionnaires. Our chatbot currently works for questions typed in by the user and expanded this work so that it can respond to voice messages as well. For further enhancements, expression recognizer can improve to detect more than just the 4 basic emotions that it currently recognizes.

REFERENCES

[1] Yadav, U. Gewali, S. Khatri, S. R. Rauniyar and A. Shakya, "Smart Job Recruitment Automation: Bridging Industry and University," 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 2019, pp. 1-6, doi: 10.1109/AITB48515.2019.8947445.

[2] E. Bagarukayo and E. Mwesigwa, "'Jobs256' Mobile app linking job seekers to job opportunities," 2017 IST-Africa Week Conference (IST-Africa), Windhoek, 2017, pp. 1-7, doi: 10.23919/ISTAFRICA.2017.8102392.

[3] Mohamed, W. Bagawathinathan, U. Iqbal, S. Shamrath and A. Jayakody, "Smart Talents Recruiter - Resume Ranking and Recommendation System," 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 2018, pp. 1-5, doi: 10.1109/ICIAFS.2018.8913392.

[4] Bing Liu, Gokhan Tur, Dilak Hakkani-Tur, Pararth Shah, Larry Heck," Dialogue Learning with Human Teaching in End-to-End Trainable Task-Oriented Dialogue Systems", NAACL, April-2018.

[5] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): n. pag. Crossref. Web.

[6] Sumit Maheshwari, Abhishek Sainani, P. Krishna Reddy," An Approach to extract special skills to improve the performance of resume selection", DNIS2010-2010.

[7] Kun Yu, Gang Guan, Ming Zho, "Resume information extraction with cascaded hybrid mode". 43rd Annual Meeting of the Association for Computation Linguistics (ACL)2005.

[8] Octavio Arriaga, Paul Ploger, Matias Valdenegro." Realtime CNN for emotion and gender classification", arXiv:1710.07557-2018.

[9] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning", IEEE-2018.

[10] A. B. Abdul Qayyum, A. Arefeen and C. Shahnaz, "Convolutional Neural Network (CNN) Based Speech-Emotion Recognition," 2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON), Dhaka, Bangladesh, 2019, pp. 122-125, doi: 10.1109/SPICSCON48833.2019.9065172.

[11] J. Wang and Z. Han, "Research on Speech Emotion Recognition Technology based on Deep and Shallow Neural Network," 2019 Chinese Control Conference (CCC), Guangzhou, China, 2019, pp. 3555-3558, doi: 10.23919/ChiCC.2019.8866568.

[12] Z. Han, S. Lun and J. Wang, "A Study on Speech Emotion Recognition Based on CCBC and Neural Network," 2012

[13] International Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp. 144-147, doi: 10.1109/ICCSE.2012.128.

[14] Sungheetha, Akey, and Rajesh Sharma. "TransCapsule Model for Sentiment Classification." Journal of Artificial Intelligence 2, no. 03 (2020): 163-169.

[15] Suma, V. "Computer Vision for Human-Machine Interaction-Review." Journal of trends in Computer Science and Smart technology (TCSST) 1, no. 02 (2019): 131-139.