



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,
SCHOOL OF ENGINEERING AND TECHNOLOGY,
SHARDA UNIVERSITY, GREATER NOIDA**

A NOVEL CV BASED CANDIDATE SUITABILITY ASSESSMENT MODEL

*A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Technology in Computer Science and Engineering*

by

Saumy Raj (2018010226)

Kartik Rathi (2018008330)

Yash Vardhan Singh (2018010015)

Muskaan Vashishtha (1801011522)

Supervised by:

Sudhir Mohan, Associate Professor

May, 2022

CERTIFICATE

This is to certify that the report entitled “ THE CANDIDATE SUITABILITY ASSESSMENT MODEL” submitted by Mr. Saumy Raj (2018010226), Mr. Kartik Rathi (2018008330), Mr. Yash Vardhan Singh (2018010015) and Ms. Muskaan Vashishtha (1801011522) to Sharda University, towards the fulfillment of requirements of the degree of Bachelor of Technology is record of bonafide final year Project work carried out by him/her in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University. The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for award of any other Degree/Diploma.

Signature of Supervisor

Prof. Sudhir Mohan

Associate Professor

Signature of Head of Department

Dr. Nitin Rakesh

(Office seal)

Place:

Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honored to have so many wonderful people lead us through in completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

My grateful thanks to Prof. Sudhir Mohan for his guidance in my project work. Prof. Sudhir Mohan, who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

Saamy Raj (2018010226)

Kartik Rathi (2018008330)

Yash Vardan Singh (2018010015)

Muskaan Vashishtha (2018011522)

LIST OF FIGURES

Figure no.	Title	Page no.
Fig 3.1	Work Flow Diagram for proposed model	22
Fig 3.2	Activity Scheduling Diagram for proposed model	23
Fig 3.3	Use Case Diagram for proposed model	24
Fig 3.4	ER Diagram for proposed model	25
Fig 3.5	DFD for proposed model (Level 0)	26
Fig 3.6	DFD for proposed model (Level 1)	27
Fig 3.7	DFD for proposed model (Level 2)	28
Fig 3.8	UI Template for proposed	29
Fig 3.9	UI Template for login	30
Fig 3.10	UI Template for registration	30
Fig 3.11	UI Template for main page Code snippet for skills	31
Fig 3.12	UI Template for submitting CV	31
Fig 3.13	UI Template for submitting / managing JD's	32
Fig 3.14	Database for registered students	33
Fig 3.15	Database for JD's and students enrolled in them	33
Fig 4.1	Code snippet for loading spaCy model	35
Fig 4.2	Code snippet for Entity Ruler	35
Fig 4.3	Code snippet for Skills	36
Fig 4.4	Code snippet for Job Description	37
Fig 4.5	Code snippet for Entity Recognition	37
Fig 4.6	Code snippet for Custom Entity Recognition	38
Fig 4.7	Code snippet for Resume Analysis	38

Fig 4.8	Code snippet for Matching Score	39
Fig 4.9	Code snippet for loading Dataset	40
Fig 5.1	Code snippet for ROUGE score	43

LIST OF TABLES

Table no.	Title	Page no.
1.1	Gantt Chart	3
2.1	Literature Review	4 - 8

ABSTRACT

Text summarization comes under the domain of Natural Language Processing (NLP), which entails replacing a long, precise and concise text with a shorter, precise and concise one. Manual text summarizing takes a lot of time, effort and money and it's even unfeasible when there's a lot of text. Much research has been conducted since the 1950s and researchers are still developing Automatic Text Summarisation (ATS) systems. In the past few years, lots of text summarization algorithms and approaches have been created. In most cases, summarization algorithms simply turn the input text into a collection of vectors or tokens. The basic objective of this research is to review the different strategies used for text summarizing. There are three types of ATS approaches, namely: Extractive text summarization approach, Abstractive text summarization approach and Hybrid text summarization approach. The first method chooses the relevant statements out of the given input text or document & convolves those statements to create the final output as summary. The second method converts the input document into an intermedial representation before generating a summary containing phrases that differ from the originals. Both the extractive and abstractive processes are used in the hybrid method. Despite all of the methodologies presented, the produced summaries still lag behind human-authored summaries. By addressing the various components of ATS approaches, methodologies, techniques, datasets, assessment methods and future research goals, this study provides a thorough review for researchers and novices in the field of NLP.

Table of Contents

Title Page	i
Certificate	ii
Acknowledgment	iii
List of Figures	iv
List of Tables	v-vi
Abstract	vii
Contents	ix

CONTENTS

Chapter-1: Introduction	1
1.1: Problem definition.....	1
1.2: Project Overview/Specification.....	2
1.3: Hardware Specification.....	3
1.4: Software Specification.....	3
1.5: Gantt chart.....	3
Chapter-2: Literature Survey.....	4
2.1: Existing System.....	9
2.2: Proposed System.....	17
2.3: Feasibility Study.....	17
Chapter-3: System Analysis and Design.....	19
3.1: Requirement Specification.....	21
3.2: System Architecture.....	21
3.3: System Design.....	28
Chapter-4: Implementation & Dataset.....	35
4.1: System modules and flow of implementations.....	39
4.2: Dataset.....	
Chapter-5: Results & Testing.....	41
5.1: Result	41
5.2: Testing.....	41
5.2.1: Type of testing adapted.....	41
5.2.2: Test results of various stages.....	43
5.2.3: Conclusion of Testing.....	44
Chapter-6: Conclusion & Future Improvements.....	45
6.1: Conclusion.....	45
6.2: Scope of Improvement	46

References

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

In today's competitive world, it is really important for Hiring Managers to choose the right and the perfect candidate for a job/work profile since students are highly competitive, the job market is less as compared to the population and various other factors. The company needs a stable and excellent candidate who helps the organization to grow and doesn't pull them behind. So, this choice becomes very important to look up for the best candidate from so many Curriculum Vitae.

For now, it is done by humans themselves, but wherever humans are included, nothing can be 100% accurate because we are bound to make errors. Although humans help to recall different factors like organizational necessities, time, attempt, and financial resources. But the human brain has some barriers, therefore, everything created, designed, or developed through humans too has some or other constraints.

There are some problems need to be looked upon when HRs select candidates:

- **Time-consuming process:** Hiring professionals are also people and need to go through so many CVs manually and read through their details. This makes them tired and hence the process becomes quite cumbersome. After so much work, the selection becomes prone to errors.
- **Unorganized CVs:** Generally, the Curriculum Vitae sent by candidates don't follow a general structure and have different formats. So the same information is present at different places in different CVs which again makes it tough to make summaries of CVs.

- **Biasedness of HRs:** Many times, employees come up with their references and refer their candidates for some profile and the HR can act biased towards those candidates and shortlist them for the profile which is wrong for the person who is more prepared for the same profile.

All these problems can be a setback for people who work so hard and send their CVs every single day to so many companies and still don't get shortlisted although they are capable of that role.

Automated systems can help us avoid all these things and can provide us with a centralized format to help the HRs with the help of an AI ML model to choose the candidates without any biases/problems based on the requirements of the company with no errors.

1.2 Project Overview / Specification

The basic idea behind this project is to save the time of the organization and also the human resources by completely making the whole process of recruitment online with the help of the ML models.

Project objective- At the end of this project, we are going to have a web-based application that will select the best CV of those candidates who are really good and right in terms of the work of the organization without any unfair means.

Project perimeter- This project is limited up to CV analysis, summarization, sorting, and in addition with personality prediction.

Project planning- This is a web-based application so firstly we will start with the frontend development of this project, then we will go for the ML model and algorithm with the database to store the CV's and then we will move forward to data training.

1.3 Hardware Specification

Hardware Components required in this software are -

- The minimum requirement of Processor is intel-i3
- The minimum storage of Hard Disk should be 500 GB
- Memory for the computational work should be minimum of 2GB RAM
- Stable Internet Connection

1.4 Software Requirement

- Windows 7 or higher.
- SQL 2008
- Visual Studio 2019
- Python v3 or above

1.5 GANTT CHART

Table 1.1 Gantt chart

Name	System ID	Workload
Saamy Raj	2018010226	Problem Identification and R&D, Code Implementation
Kartik Rathi	2018008330	Problem Identification and R&D, Frontend
Yashvardhan Singh	2018010015	Testing and debugging
Muskaan Vashishtha	2018011522	

CHAPTER 2

LITERATURE SURVEY

Table 2.1 Literature Review

Sn o.	Author	Objective/Topics Focused on	Algorithm/Model s/Framework used	Accuracy /Results	Conclusion
1	Lino Mathew/Nikitha Linet /Nithin C George/Nithin K Thomas [1]	This paper proposes a model that computes a comparison and based on the results, suggestions are provided to the candidates to modify their resume. The proposed procedure extract information related to technical as well soft skills from the CV's submitted in text, pdf or docx format. The system also provides suggestions for correcting the grammatical errors. The proposed system is designed based on Natural Language Processing (NLP) techniques.	NLP(Spacy Library),NER/React JS,Flask,PostgreSQL	NA	A different way of evaluating and analyzing the data in a CV/Resume is proposed in this system. This is because the system helps the candidates to have prior knowledge about to what extent they have the chances of clearing the screening process based on their resume. The proposed system extracts technical information from the CV and categorizes them for comparison. It also successfully store the analyzed results, which the user can refer to for future purpose.
2	Pradeep Kumar Roy/Sarabjeet Singh	Being able to weed out non-relevant profiles as early as possible in the pipeline results in cost	Classifier-Random Forest,Multinomial Naive Bayes,	RF-0.3899, MNB-0.4439,	The process of classifying the candidate's resume is manual,

	Chowdhary/ Rocky Bhatia [2]	savings, both in terms of time as well as money	Logistic Regression, Linear support Vector Classifier / Content Based Recommendation Cosine Similarity and k Nearest Neighbours	LR-0.6240, LSCV-0.7853	time consuming, and waste of resources. To overcome this issue, they have proposed an automated machine learning based model which recommends suitable candidate's resume to the HR based on given job description. The proposed model worked in two phases: first, classify the resume into different categories. Second, recommends resume based on the similarity index with the given job description. The proposed approach effectively captures the resume insights, their semantics and yielded an accuracy of 78.53% with LinearSVM classifier.
3	Vikas Yadav, Steven Bethard [3]	It presents a comprehensive survey of recent advances in named entity recognition. It describes knowledge-based and feature-engineered NER systems that combine in-domain knowledge, gazetteers, orthographic and other features with supervised or semi-supervised learning. It contrasts these systems	NA	First finding from the survey is that feature-inferring NN systems outperform feature-engineered systems,	Our survey of models for named entity recognition, covering both classic feature-engineered machine learning models, and modern feature-inferring neural network models has yielded several important insights. Neural network models generally outperform feature-

		with neural network architectures for NER based on minimal feature engineering, and compare amongst the neural models with different representations of words and sub-word units.		despite the latter's access to domain specific rules, knowledge, features, and lexicons. The next finding is that word+character hybrid models are generally better than both word-based and character-based models.	engineered models, character+word hybrid neural networks generally outperform other representational choices, and further improvements are available by applying past insights to current neural network models, as shown by the state-of-the-art performance of our proposed affix-based extension of character+word hybrid models.
4	Yiou Lin, Hang Lei, Prince Clement Addo, and Xiaoyu Li [4]	In this work, they evaluate the job matching problem as a classification problem. This is to identify a job seeker's current employment detail (the last position in the resume) by their previous employment history	keras,sklearn/kmeans cluster/Random Forest and Extreme Gradient Boosting(XGB)/CNN,LSTM	XGB performs best among four baseline estimators with longest training time(53m 19s), while CNN model converges in shortest	In this paper, they have considered the resume-job matching problem and proposed a solution by using unsupervised feature extraction, supervised machine learning methods and ensemble methods. The solution is completely date-driven and can detect similar position without

				time(1m 14s) with acceptable precision	extra semantic tools. Besides, our solution is modularized and can rapidly run on GPU or simultaneously run on CPU. Compared to a manual rule-based solution, our method shows better performance in both precision and Top-N recall.
5	Thomas Schmitt, Philippe Caillou, Michèle Sebag [5]	This paper shows that the information inferred from their (Candidates and recruiter) interactions differs from the information contained in the CVs and job announcements. The second contribution is the hybrid system Majore (MAtching JObs and REsumes), where a deep neural net is trained to match the collaborative filtering representation properties	Deep Neural Network with Collaborating Filtering		A promising approach is presented in this paper, using a deep neural network to emulate the metric properties of the oracle, collaborative filtering-based, representation. More complex architectures (e.g., handling the geographic information; considering domain adaptation among the job seeker and the recruiter spaces) will be considered in further work. A main challenge will be to adapt to the evolution of the actual user behaviours, responding to the evolution of the job matching platform.
6	VINAYA RAMESH KUDATARKAR, MANJULA	The main goal of this work is extracting the resume information which makes the job	Resume Parser, Concept Builder, Analysis		The work contributes to the discussion presenting a resume

	RAMANNAV AR, DR. NANDINI S.SIDNAL [6]	easier by finding the suitable resume to fit their needs.			parser in which the grammar and probabilistic parameters are induced from a tree bank and have shown that its performance is superior to previous parsers in this area. An experiment that suggests that its superiority stems mainly from unsupervised learning plus the more extensive collection of statistics that it uses, both more and less detailed than those in previous systems. The proposed model collects resumes through web search and ranks based on cosine similarity measure. Statistical parsing plays vital role while extracting and keeping the information relevant and up-to-date. Thus the search time for required document is reduced when data is stored. The model also reduces the human effort required in seeking the relevant information
--	--	---	--	--	--

2.1 Existing Systems

1. Summarization using Pretrained encoders- BERT (Bidirectional Encoder Representations from Transformers) [7] is a new technique for pretrained language models that has gradually gained a wide spectrum of natural language processing (NLP) jobs. BERT can propose a framework and architecture for both extractive and abstractive summarization. It's essentially a novel language representation model that uses masked language modelling to train.

Extractive - A neural encoder generates sentence representations, after which a classifier predicts which sentences should be chosen as summaries, rearranges them, and finally adds the necessary grammar. Different models like REFRESH [8](it is a reinforcement learning-based system that has been taught by maximizing the ROUGE measure worldwide.), LATENT [9](Given a set of phrases, this latent model maximizes the likelihood of human summaries.), SUMO [10](It basically uses structured attention to instigate or provide a multi-root dependency tree representation of the material while anticipating the output summary.), NEUSUM [15](it is the sophisticated extractive summarization scores and chooses sentences together.) have been used for extractive summarization.

Abstractive - The job is viewed as a sequence-to-sequence challenge. Different models like PTGEN((pointer generator network)[12] It has a word copying feature that allows it to copy information from the original input, as well as a coverage feature that maintains track of terms that have been summarized.), DCA ((Deep Communicating Agents)[13] models are trained with the help of reinforcement learning), DRM (deep reinforced model)[14] for abstractive summarization, which solves the coverage problem by employing an intra-attention technique in which the decoder pays attention to previously created words.) Are currently being used for abstractive summarization.

Dataset - Three separate benchmark datasets were used to test all of these models. Sum, the CNN/Daily Mail news highlights dataset, the New York Times Annotated Corpus, and the CNN/Daily Mail news highlights dataset [15]. These datasets cover a variety of summary styles or patterns, from highlights to one-sentence summaries for input or testing.

Implementation - They mostly employed PyTorch for both extractive and abstractive summarization scenarios. [20], to implement BERTSUM, use OpenNMT and BERT's 'bert-

base-uncased' version. BERT's sub words tokenizer was used to tokenize both the source and destination texts.

2. NER Summarization - Named Entity Recognition (NER) [41] is a method for identifying and categorizing atomic things in text into predefined categories, such as people's names, organization names, places, concepts, and so on. NER is now employed in a variety of applications such as text summarization, text categorization, question-answering, and machine translation systems in a variety of languages. There has already been a lot of work done in the subject of NER for English, where capitalization provides a crucial indication for rules, however Indian languages lack such qualities. This complicates the process of summarizing the material in Indian languages.

Relation Extraction - It is the task of identifying and locating the semantic connections between things in text texts, and it is another essential information extraction activity. How to categorize the relationship between two things into one of the fixed connection categories given a couple or group of entities co-occurring in a phrase. Given a pair or set of items that appear in a sentence together, how to classify a relationship between two entities as one of the fixed relation types. Although relations can span numerous sentences, this is uncommon, therefore most previous research has concentrated on relation extraction inside the phrase. There have been several research that have used the classification technique to extract relationships.

NER Frameworks

Stanford CoreNLP - Stanford CoreNLP [16] is a Java-based natural language processing framework that can handle a wide range of jobs. In Stanford CoreNLP, the CRF model is employed for NER. The user can select which characteristics should be used to attempt to forecast named things, however embedding words are not accepted.

SpaCy - SpaCy [17] is a Python-based open-source library used in natural language processing. It has a very well, simple API, is speedy, and has pre-built neural network models that are "good" for most cases with few parameters that need fine-tuning, making it simple to apply to real-world applications. The difficulty in this library is that it can't be used to do research because

there aren't more factor to play with & the neural network designs can't be changed. It's also a drawback because SpaCy only supports static word embedding's like FastText and GLoVe.

Flair - Flair a, straightforward framework designed by Zalando Research to accomplish cutting-edge NLP. Flair has its own NER architecture, which includes a bidirectional LSTM and a CRF decoder. This framework has the advantage of supporting all common word embedding's, from GLoVe to the most recent contextual embedding's like Bert and Elmo. Flair is a set of contextual embedding's proposed by Akbik et al. [18]. Users of this library may easily experiment with mixing various embedding's without altering the neural network architecture repeatedly, as they would if they used a machine learning framework such as TensorFlow [19] or PyTorch [20].

3. Sequence to sequence RNN Summarization - This concept enables sequences from a single domain to be changed into sequences from another domain. They began by describing the basic encoder-decoder RNN, which serves as a baseline, before presenting a variety of novel summarization models. Neural machine translation model is being depicted by this baseline model. The bidirectional GRU-RNN is being used by the encoder, whilst the unidirectional GRU-RNN is being used by the decoder with a encoder as the same hidden-state size and words are produced using attention to the tool over the source, for example: the hidden states and a soft-max layer gets more attention over the target vocabulary [21, 22] .

The summarizing problem, the huge vocabulary 'trick' (LVT), was also adjusted or added to this core model. This method's major goal is to lessen the size of data of the decoder's softmax layer, which is the main computational bottleneck. Furthermore, by limiting the modeling effort just on those words which are crucial with respect to a specific example, by following this type of strategy speeds up convergence. Because a major part of the words in the summary originate from the original material, this approach is excellent for summarizing [23].

Dataset - The following datasets were employed in all of these models: Gigaword Corpus, CNN/Daily Mail Corpus and DUC Corpus,

Implementation - In addition, each highlight was treated as a single sentence between system and gold summary, as opposed to the full-length Rouge F1 metric used for the Gigaword corpus.

Results - On this dataset, both RNN [21] and hierarchical models produced summaries that contained repetitious words or even repetitious sentences at times, according to a visual evaluation of the system output. Because the summaries on this dataset comprise many phrases, it is feasible that the decoder 'forgets' which a part of the record turned into used to generate preceding highlights for this approach. To solve this difficulty, they employed the Temporal Attention model, which maintains account of the decoder's previous attentional weights and expressly forbids it from paying to comparable sections of the document in subsequent steps in terms of time.

4. Bayesian Learning - SUMARIST, SWESUM, and other automated text summary systems have been developed for the English language. However, a single-syllable [24] languages such as Vietnamese, Chinese, Japanese, Mongolian, Thai, and other "native" languages of East Asia and Southeast. Many people speak single-syllable languages, which account for more than 60% of all languages spoken on the planet. As a result, processing a one syllable language is critical. However, it is quite difficult to detect a word or phrase solely on white space, and all word segmentation techniques presently do not achieve 100% accuracy. They primarily suggested a text summary approach based on the Nave Bayes algorithm [25] and a subject word set in this research report.

For single syllable text, Nave Bayes classification is utilized in two phases: Training and summarization are two important aspects of the job. We trained using data and with the help of people to create a collection of extracted sentences in the Training phase.

Dataset - The authors employed Vietnamese text in their studies, and they also constructed an automated text summarizing system to handle text summarization efficiently and simply using the suggested techniques. A Vietnamese text corpus has been constructed for the purpose of conducting summary experiments on Vietnamese texts.

5. Fuzzy logic - It is a typical model based on fuzzy logic for Automatic Text Summarization and it takes eight features as the input for each and every sentence like (Length of sentence, Data in numerical form, Location of a sentence, Title word, Thematic words, Sentence to sentence similarity, Proper noun and Term weight) for its basic importance calculation. After

extracting these eight features attributes values, it goes into a Fuzzy Inference System (FIS). Also, according to the indagation, a summary length of roughly 10% (approximately) of the real text length is appropriate and the resulting summary consists of phrases extracted in the original sequence. The Main components of FIS are:

Fuzzification – To convert the crisp values to fuzzy values, at this stage we will use the membership function. Many membership functions are available for mapping, including triangular, trapezoidal, ball, and Gaussian distributions.

Inference Logic - With the input value obtained in the first stage, an IF-THEN knowledge base is constructed, and with the help of inference engine output is generated based on these rules. The weights of important and non-key components are balanced using IF-THEN rules.

IF-THEN – this rule is stated in the following format - If title similarity is moderate, sentence position is moderate, sentence length is moderate & numerical data is high, medium or low then output is key.

Defuzzification – Last stage of fuzzy logic-based summarization, in this the membership function is used to transform the results from the second phase to crisp values, that mainly means that the linguistic result it translated into a numeric number from the inference engine. Depending on the type of scenario, the output membership function might be the same of trapezoidal or other than that. To compute the crisp value centroid method is used.

Dataset - Despite the fact that the DUC 2002 datasets are the de-facto standard data sets for testing summarization systems, only around half of research have apparently used them. Thus, this makes comparing the empirical results indistinct, indefinite and non-uniform, thus identifying the need for more benchmarked studies and subsequent evaluation.

6. Latent semantic analysis - In the context of Latent Semantic Analysis, one discovers hidden semantic structures in words or sentences using algebraic-statistical methods. It's an unsupervised method that doesn't need any training or prior knowledge [27]. LSA gathers information from the context of the input material, such as whether words are used together and which common terms appear in various phrases. Many words in the sentences are common,

which suggests that they are semantically related. In a sentence, the meaning of words is determined by the context in which they appear, and in a word's context, its meaning is determined by the context in which it appears. A mathematical approach, Singular Value Decomposition, is used to discover the relationships between phrases and words [28]. SVD [28] can predict links between words and phrases as well as reduce noise, which improves accuracy.

Step 1: Forming an input matrix: For a computer to understand and process a document, it must be formatted correctly. This is commonly represented as a matrix, with the sentences as columns and the words/characters as rows. The cells are also used to illustrate the significance of each sentence's words.

Step 2: Singular Value Decomposition (SVD) is an algebraic approach for modelling connections between words and sentences [28].

Step 3: Sentence selection: the key sentences are chosen using the SVD findings as well as various methods.

Following Sentence selection approaches have been used:

1. IR method [29]
2. Content Based method [30]
3. Gaussian Mixture Model [31]
4. Cross method [32]
5. Topic method [33]

Implementation: In this research, the LSA-based summarization techniques are evaluated using Turkish and English datasets. Different input matrix construction methods are used for different LSA methodologies. Stemming and stop word removal procedures are employed to reduce the size of the final matrix. All the summaries created have a length of 10% of the input document. The ROUGE evaluation metrics are used to underpin the evaluations.

Dataset - Turkish Dataset [34] -For the assessment of summarization methodologies, four distinct sets of Turkish papers were employed. The first two sets of papers are scientific in nature, with topics ranging from medicine to sociology to psychology. There are 50 articles in each dataset. The articles in the second dataset are significantly lengthier than those in the first.

English Dataset - The datasets utilized to assess the LSA-based summarizing technique and methods are the Duc2004, Duc2002, and Summac datasets. For single-document summarization, all datasets are utilized. Some tasks in the Duc datasets have been specified to limit the output summary size according on the requirements. In Duc datasets, the output sizes are extremely small, e.g.100 words or fewer, which might impact or restrict the quality of the derived summaries.

7. MS Pointer Network – After a period of time, QianGu using the so-known Multi Source-Pointer technique is the next analysis received from the ML approach. This technique primarily focuses on assigning a rating to abstractive using deep learning by predicting the inaccuracy of words in the text as well as semantic inaccuracy. Basically, in this term, larger weights are assigned to words that are semantically related. The rogue is tested on the Gigaword and cable news network (CNN) datasets for this method's assessment. In compared to other ML techniques such as Sequence to sequence in addition to attention baseline, as well as Nallapati's abstractive model and the results performed quite well. The Gigaword dataset was used to test this model and it was superior to rouge-1 scoring 40.21 as shown to be, rouge-2 scoring 19.37 and rouge-L scoring 38.29. Another test is conducted using the CNN dataset, with rouge-1 scoring 39.93, rouge-2 scoring 18.21 and rouge-L scoring 37.39. Other than all this methods, another one is loses rouge-1 scoring measurements, which is basically contrasted with systems like baseline lead-3 given by Nallapati, with rouge-1 scoring 40.21 in the dataset of CNN. The major disadvantage of this type of model is the occurrence of recursion of the same statements in the document. By the virtue of this, it can be seen that this type of model is mainly kindred to the recursion/redundancy issue of the sentences. Qian Guo, a problem researcher, suggests adding TF-IDF or RBM to achieve a suitable or correct summaries which results in context of future study [35-37].

8. Rule Based – In the last ten years, this strategy has become much less prominent in the field of text summarization. The approach's key benefit is that it can be used to a basic domain, making rule-based validation relatively straightforward. However, when utilized for a domain with a level of complexity very high, rule-based validation becomes quite difficult, so if the system is not able to identify the rules, then it cannot produce results. Aside from that, if there are more rules than are necessary, the system finds it challenging to sustain the output's performance [38].

9. Maximal marginal importance (MMI) – Current and recent ML technology studies include the Maximal Marginal Importance (MMI) approach, the PSO and a combination of other strategies such as fuzzy logic. Input is one type of document and the output is in extractive summary format. MMI produces summaries that sum up differently by determining the most unique sentences. Key sentences are chosen by taking the repetitive sentences there in the input and also by removing statements from the given input or from text-source. Techniques like PSO are used to select the least & most essential features and this fuzzy logic helps it to determine the values for the factors such as risk and ambiguity or the endurance rate can easily fluctuate. Output was then tested and verified on database of Document Understanding Conference -2002 and then compared with different types of summaries like Sys-19, Sys-30 and MsWord summaries. Results performed more than expected with the comparison with the terms which are recall scoring 0.40 and f-measure scoring 0.422. MMI, PSO, Fuzzy are superior to different summaries like Sys-30 by the accuracy of 0.063. The main disadvantage of this method is the issue of semantic problems. This approach may be used by labelling the semantic roles in the lexical dataset and other for multi-document summarizers [39, 40].

10. TF-IDF Technique – TF-IDF approach is used in text summarizing research such as [42-45]. This is from one of the algorithms that checks the link between a text and the entire collection of documents available. The major goal here is to compute the TF and IDF values. Every phrase is treated as a separate document for a single input or a single type of document. The frequentness of recurrence of the word (T) in the entire single statement is used in this approach to determine how essential that word is in the input. IDF, on the other hand, is a numerical figure that represents the frequentness of the term (T) appears in a sentence. The

numerical value or weightage of the word will be much more if it occurs many times in the document and also least in many other documents, one can find this by simply multiplying the TF value to the IDF value [46].

2.2 Proposed System

From all the above models it is concluded that with the use of NER method, we can provide systematic, comprehensive, clear and wide review, from the field of styles/topics, data sets, different coping strategies, problems, available assessment methods as a guide to future work. NER works best on the group of resumes as it works for extractive summarization so, it summarizes the given resume according to the words present in the resume without altering or changing them which makes it more precise for HR's point of view to see the summarized document.

The important factor that is considered as interesting in the analysis of results, which says that in comparison to the abstractive summaries extractive summaries are much simpler, extractive summaries are still the subject of present popular trends. This is because more research need to be done and also many things are still left to unravel in the abstractive summarization which is a challenge that researchers have to deal with. It can also be seen that the most essential factors in producing a good or clean summary are semantics, similarity, sentence length, sentence location, frequency, keywords and need to be there.

2.3 Feasibility Study

A CV is a type of document through which the candidate/employee is shortlisted or selected for a profile of the job. It takes longer to go through all the CVs and decide the better candidate.

Here we are with an idea to build a web-based application through which the personality (suitability) can be accessed and the hiring process becomes easy.

The user interface will be simple and easy to use by a common man. The HR department can use this application in shortlisting the right candidate with great ease. Based on skills and experience the prediction model will shortlist the candidate. The feature of the CV based employee suitability model are as follows:

- It is a user-friendly web-based application.
- With respect to the business sector, this application can be used to get the right person for the particular position.
- It is compatible with a normal desktop/laptop which has a web browser installed in it and has internet access.
- The workload of the human resource department can be reduced.
- It is economically affordable as it does not require any special requirements.
- This application will be user-friendly since the user interface will be simple and easy to use by any non-technical person.

Through the analysis, we can say that our project can be completed successfully with ease.

CHAPTER 3

SYSTEM ANALYSIS & DESIGN

3.1 Requirement Specification

3.1.1 Purpose- The human resource department's primary duty is recruiting and selection, and the recruitment process is the first step in developing a competitive and lucrative employment strategy for businesses. The recruitment process is a methodical procedure that involves a lot of resources and time, from finding candidates to organizing and performing interviews.

3.1.2 Use-Case Model Survey

The use case figures shows two people: HR and Candidates, as well as five use cases: announce the openings in the organization, complete out a referral form, choose suitable and eligible candidate, conduct an interview, and disclose the result to the candidates.

3.1.3 Use Cases

1. **Announcement of openings:** The Company HR informs about the opening details in the organization for the required job. If any opening is available, the right notification needs to be completed.
2. **Completing the referral form:** The candidate will fill out the job application form, also along with the application form, the candidate resume is also required and need to be submit. From those lists of candidates, a large number of people will apply for the job. HR will conduct a shortlisting process based on the position's needs.
3. **Conducting an interview:** HR of the Company then approaches the selected candidates for an interview process. The interview approach can be broken down into three steps: aptitude test followed by technical test, group discussion (GD), and one-to-one interview.

4. **Disclosing the results:** After the interview procedure is done, HR lists the chosen applicants and release an offer letter/ letter of intent to him/her. The candidate has a choice either to reject the offer or accept it to fill the opening of the company.

3.1.4 Assumptions and Dependencies

- The project is done for the selection process of candidates for the company
- Every candidate should be comfortable working with the computers/laptops and internet surfing.
- He/she should have a prior knowledge of the hiring system of different companies.
- The candidate should be well-versed in the English language.

3.1.5 Requirements

Below are functional and non-functional requirements.

Use-Case Specifications

1. **Announcement of openings** - The HR manager of a specific region shall inform details about the vacancy to the candidates.
 - **Pre-Condition:** Vacancy ought to exist.
 - **Post-Condition:** The job profile and vacancy are described in detail.
2. **Completing the referral form** - HR manager processes the forms filled by the candidate and selects the list of eligible candidates.
 - **Pre-Condition:** There must be an online form. (The candidate must fill in all essential information.)
 - **Post-Condition:** Forms filled are stored in the database and further used for processing.
3. **Conducting an interview** - Interviews are conducted by the HR manager of the region with the vacancy. Following the interview process, a list of selected candidates is compiled.

4. **Disclosing the results** - The candidate has been notified of their job selection. In order to fill the position, the candidate accepts the job offer. Alternatively, the candidate rejects the offer letter.

- **Pre-Condition:** The candidate is chosen for the position.
- **Post-Condition:** The candidate either accepts or declines the offer.

3.1.6 Usability

- The system shall allow users to access the system's user interface online using HTML and related technologies. The system will use a web browser as the interface as it is going to be a web-based application. It has to be considered that all the candidates are familiar with basic browser usage, so there will be no need for special training of the candidates.
- As the user interface, the system will be a web browser, since all users are familiar with how to use browsers in general, no special training is required.

3.2 System Architecture

3.2.1 Workflow Diagram

- It is a step-by-step, linear illustration of a business method from start to complete. It indicates how individual responsibilities, actions, or resources flow between diverse people or groups. It also indicates what needs to be achieved so as for that venture to be completed.
- In this diagram we will start with the login phase after that we will go for CV or Resume input from the candidate so that our CV processing model can process it

and can show the user, output in the form of a summarized CV and its generated score.

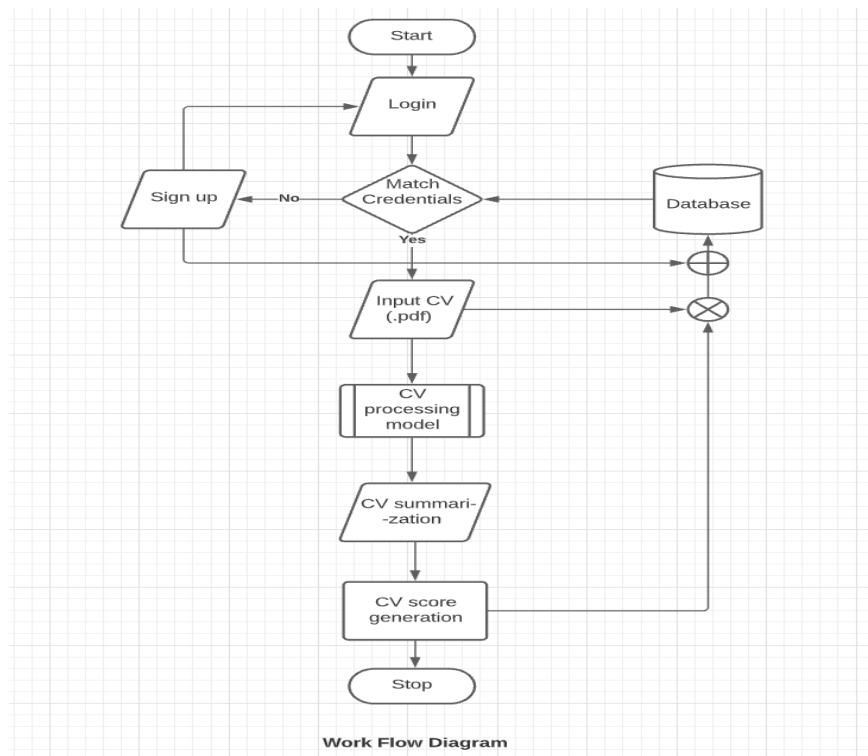


Fig 3.1- Work flow diagram for proposed model

3.2.2 Activity Scheduling Diagram

- It suggests the order wherein activities should be scheduled to address logical relationships between those activities.
- Firstly, everyone is supposed to log in/signup. In the next step, users have two options either to apply for the test and get their score calculated or upload their resume and get their CV summarized. On the other hand, HRs can post for jobs and tests for candidates to apply to. All these steps can happen simultaneously.
- Finally, the HRs can do profile shortlisting, view summarized CVs, and send emails to candidates for further process.

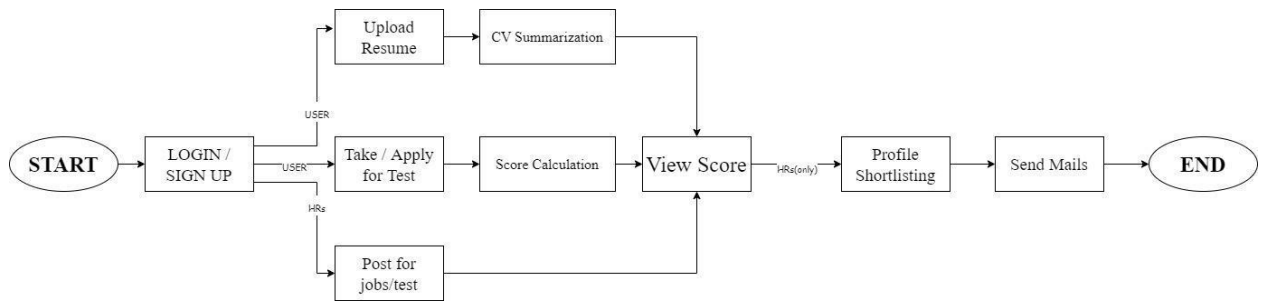


Fig 3.2- Activity Scheduling Diagram for proposed model

3.2.3 Use Case Diagram

- Describes high-level functions and scope of the system. These diagrams also recognize the interchanges between the system and its actors/characters. This diagram shows and explains the context and the requirements of the whole system or key components of the system.
- In the previous diagram, we have 3 actors – Candidates who will log in/sign up, upload the resume, and will apply for the test, second is HR who will view the score generated by the model and also shortlist the candidate's basis on that score only and the last one is the Admin who will look after the system and database management.

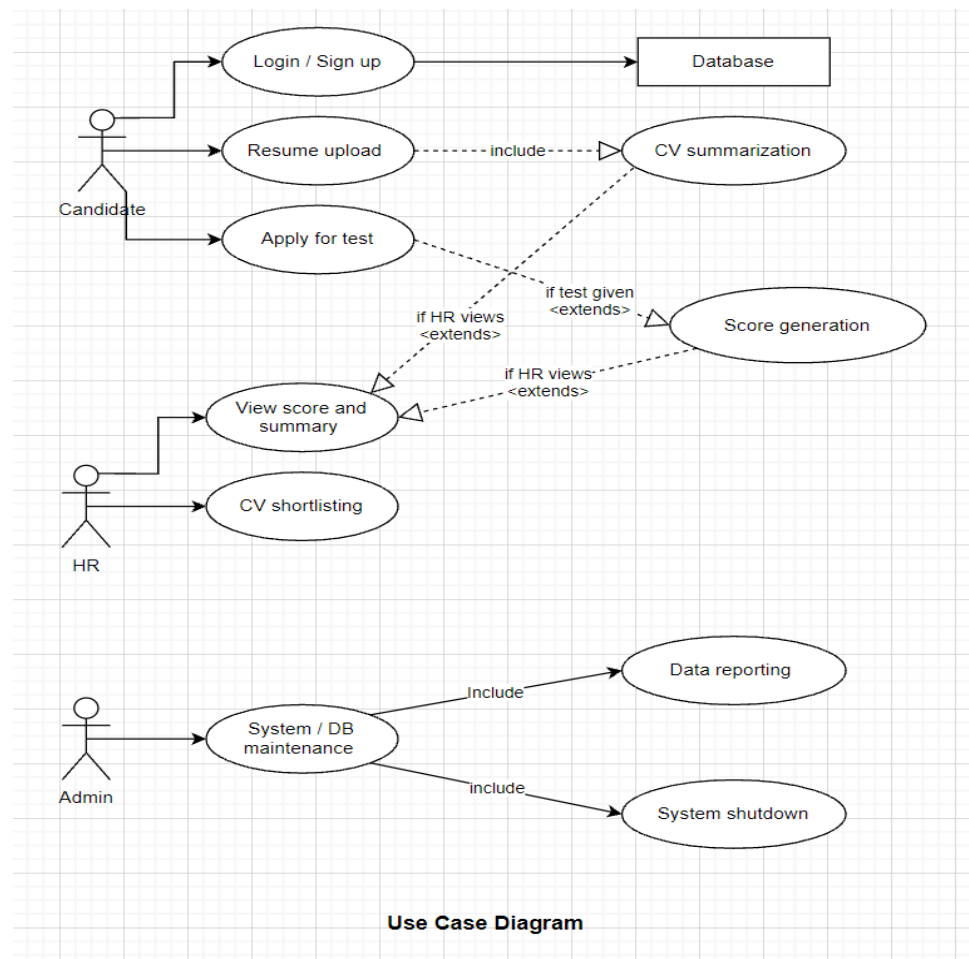


Fig 3.3- Use Case Diagram for proposed model

3.2.4 ER Diagram

- An Entity-Relationship (ER) Diagram is a sort of flow chart that shows how “entities” including people, objects, or ideas relate within a system. ER Diagrams are most often used to design or to correct information in relational database mistakes in the area of software engineering.
- The skills are matched with the keywords of the job description. The CV result generated will contain brief information such as name, address, skills, location, experience, projects, hobbies, etc.

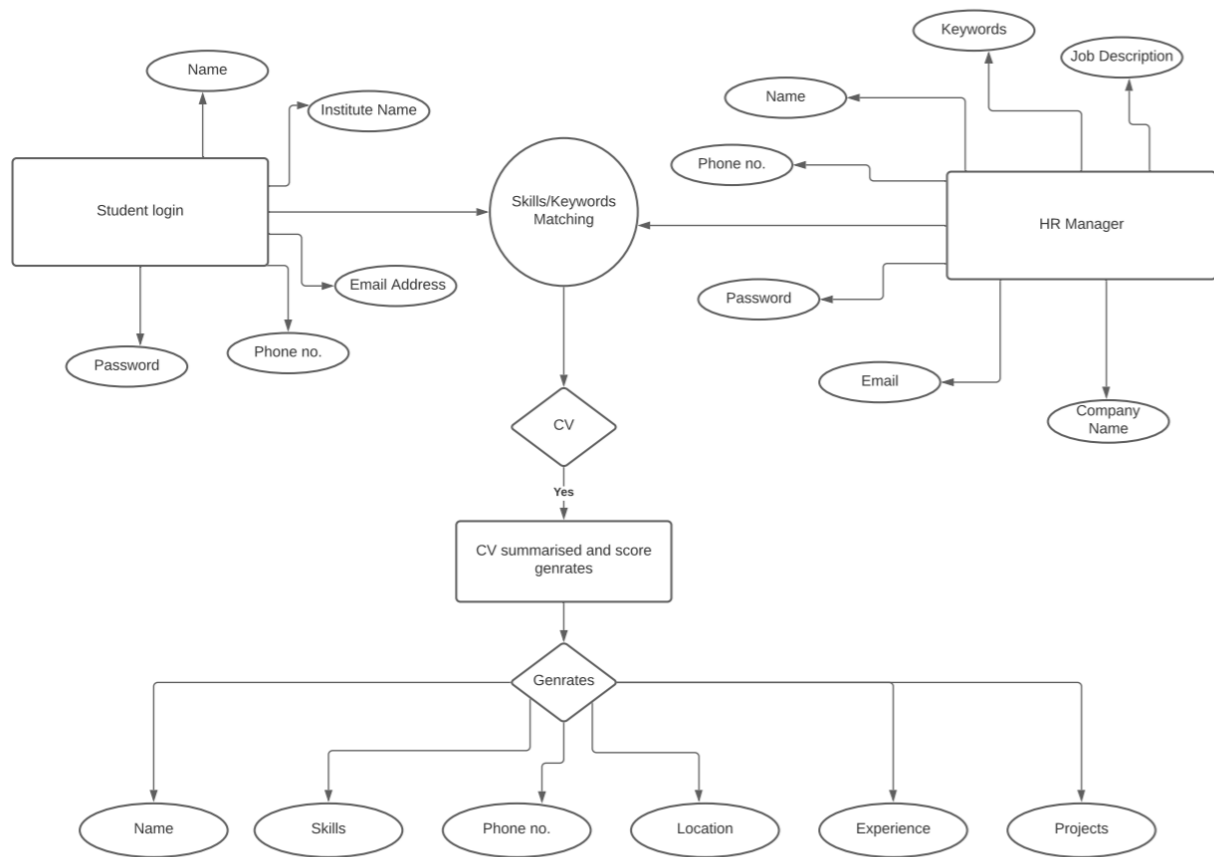


Fig 3.4- ER Diagram for proposed model

3.2.5 DFD Diagram

- The 0 Level Data Flow Diagram is likewise called a Context Diagram. It's a fundamental overview of the entire system or manner being analyzed or modeled. It is designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to outside entities.
- In this 0 level DFD we will have different models listed below:-
- Login management

- CV management
- CV test score management
- Database management
- System User management

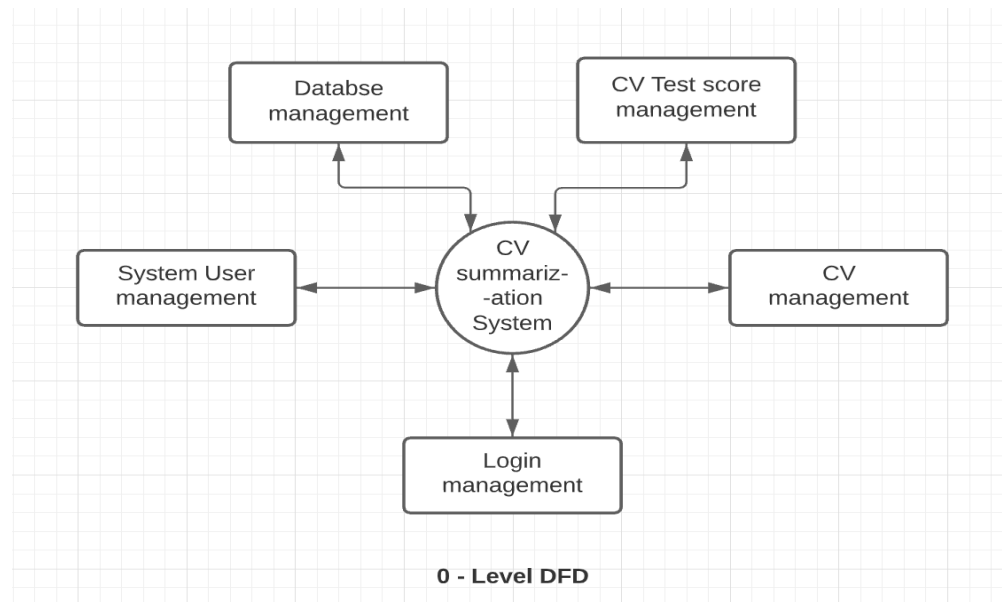


Fig 3.5- DFD Diagram (Level 0) for proposed model

Context diagrams (level 0 DFDs), as previously stated, are diagrams in which the entire structure is depicted as a single procedure. A level 1 DFD lists all of the major sub-processes that make up the overall system. A level 1 DFD can be thought of as an "exploded perspective" of the context diagram.

Following sub process we will have for our different models:-

- Login management – it will check for user login details.
- CV management – it will basically summarize the CV & manage it in DB.
- CV test score management – it will shortlist the CV on the basis of score.
- Database management – it will store all the details of candidates and resume
- System User management – it will look for multiple users and Hrs.’

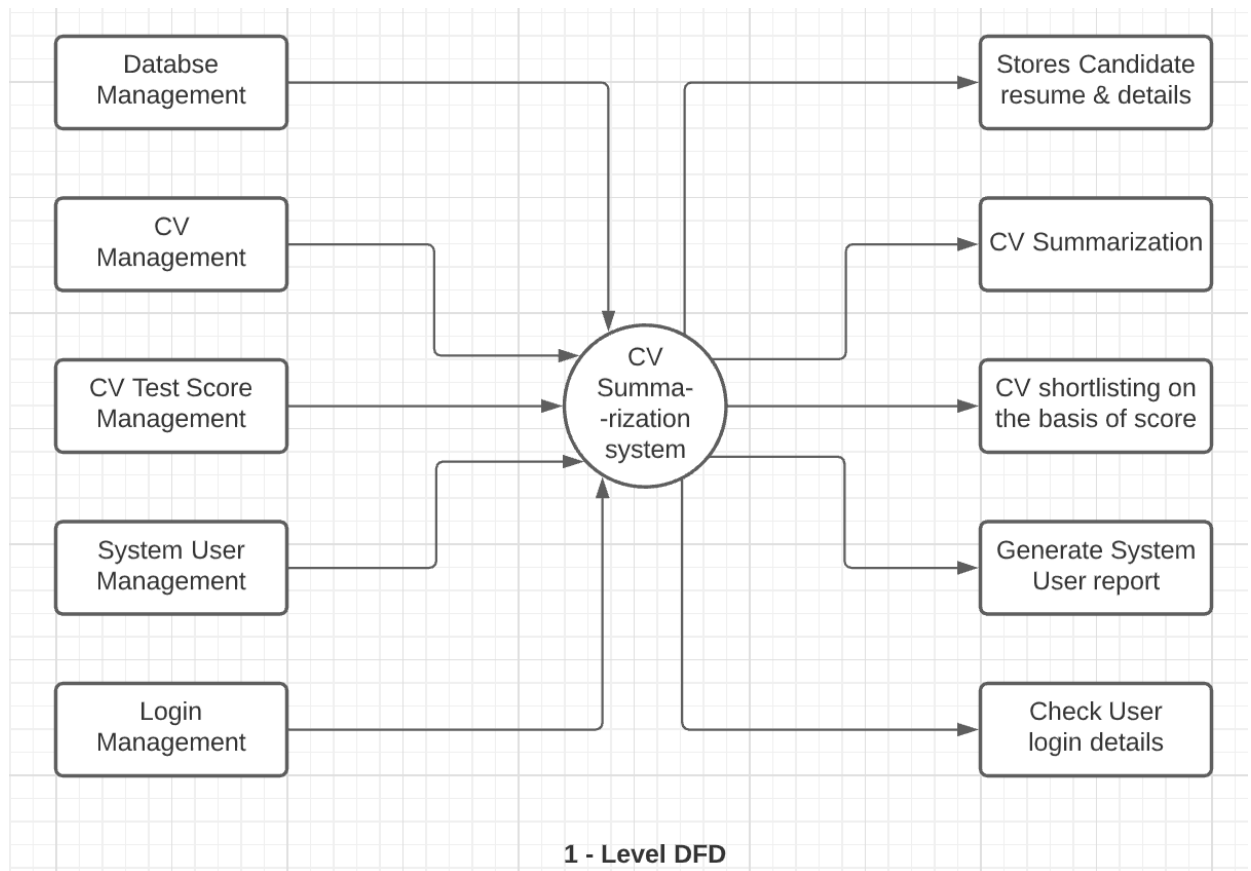


Fig 3.6- DFD Diagram (Level 1) for proposed model

The 2-level Data Flow Diagram is going one step deeper into components of 1-level DFD. It could be used to devise or report the particular/important details about the system's functioning. Or it gives an extra detailed study of the processes that make up a statistics system than a level 1 DFD does. By this admin will login into the website and then the system we check what access he/she can have for the system and according to that system will allow him/her to change or manage the different modules accordingly as given in the previous diagram.

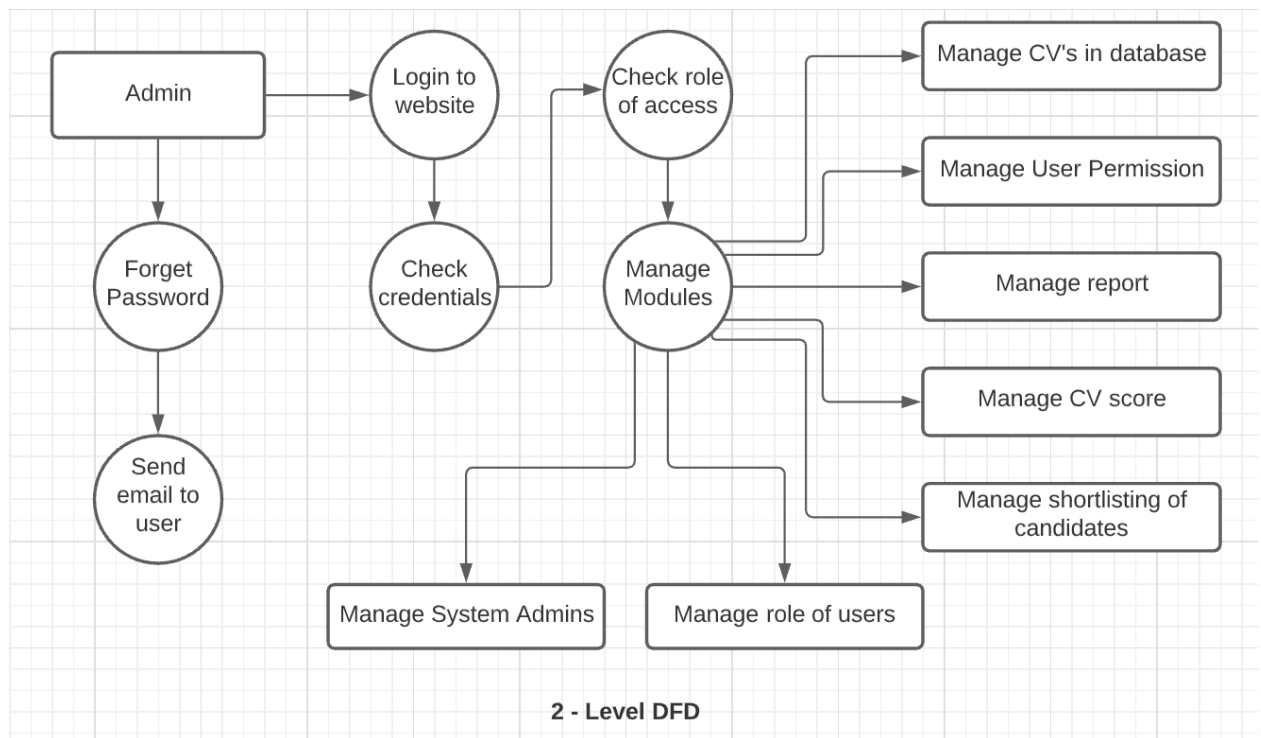


Fig 3.7- DFD Diagram (Level 2) for proposed model

3.3 System Designs

3.3.1 Design Template

CVAnalysis

[Home](#) [About](#) [Contact](#) [Login](#)

Save time, increase efficiency and boost productivity with our CV Analysis software.

The fully automated workflow solution and CV analysis software seamlessly loads candidate data from the Resume which they provide.



[About](#)

Our Project

This is a dummy project we created for the Companies to minimize their work of recruiting the right candidate from the bulk of CVs.

[read more](#)

How we work

We take CVs from the Candidates and match it according to the Job-Description(JD) provided by the company & give the result in the form of Matching Score.

"Higher the Matching Score,
Higher the chance of getting recruited"



Our Creative Team



Kartik Rathi



Saumy Raj



Yash Vardhan Singh



Muskan Vashishtha

Contact us

Your name

email address

phone no

[Submit](#)

CVAnalysis
© 2022

Fig 3.8 - UI Template for proposed

CV Analysis

Login to go onto the new page

Login

[Forgot account ?](#)

Create a new account

Create a page for your profile or business

Fig 3.9 - UI Template for login

Create a new account

Firstname

Firstname

Middlename:

Middlename

Lastname:

Lastname

ID:

Id

Gender :

☒ Male ☐ Female ☐ Other

Phone :

+91

phone no.

Current Address :

Current Address

Email

Enter Email

Upload Picture :

Choose File No file chosen

Password

Enter Password

Re-type Password

Retype Password

Submit

Fig 3.10 - UI Template for registration



Fig 3.11 - UI Template for main page

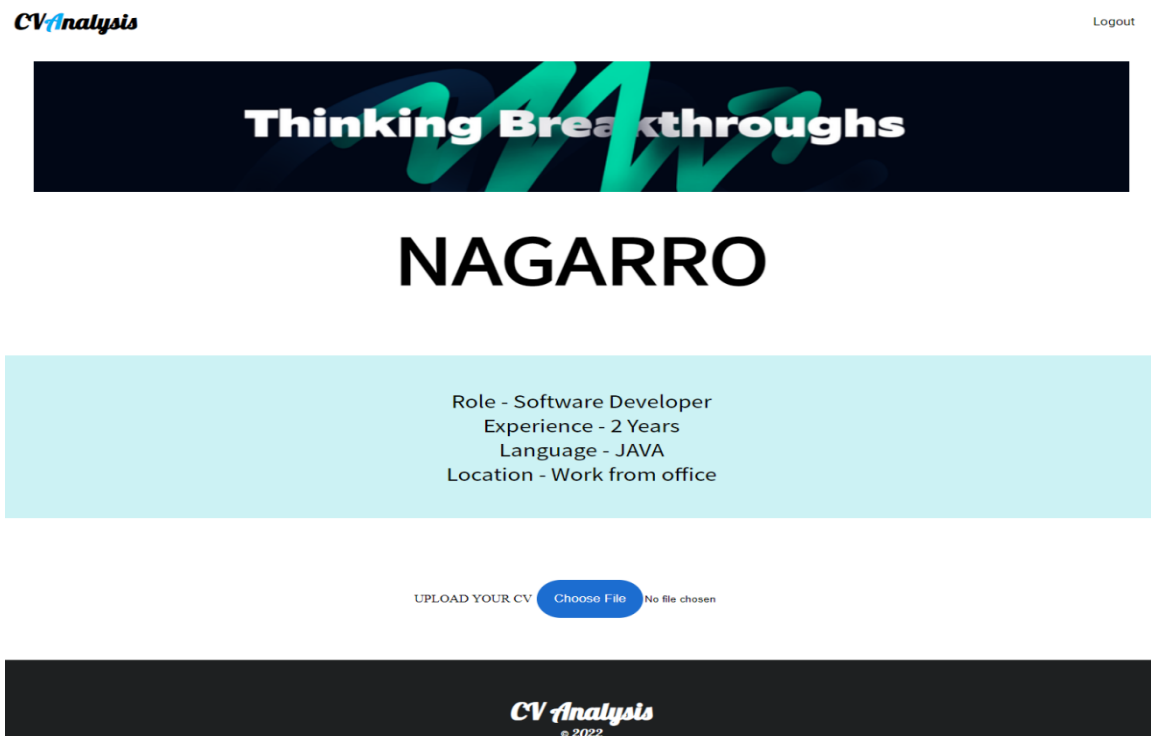
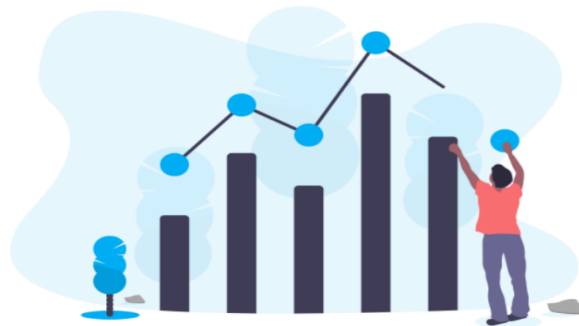


Fig 3.12 - UI Template for submitting CV

Welcome Back, Company Name
Some lines About company



Previous JD's

Your JD's

These are all the JD's and the role you provided previously. You can manage it from here by clicking on them.

#	Role	Submitted JD	Students Enrolled
1	Java Developer	blank	40
2	Python Developer	blank	30

update JD's

Upload a new JD

Upload your new JD in the given pdf or word format below.

Role

JD
 No file chosen

Contact us

Submit

Fig 3.13 - UI Template for submitting / managing JD's

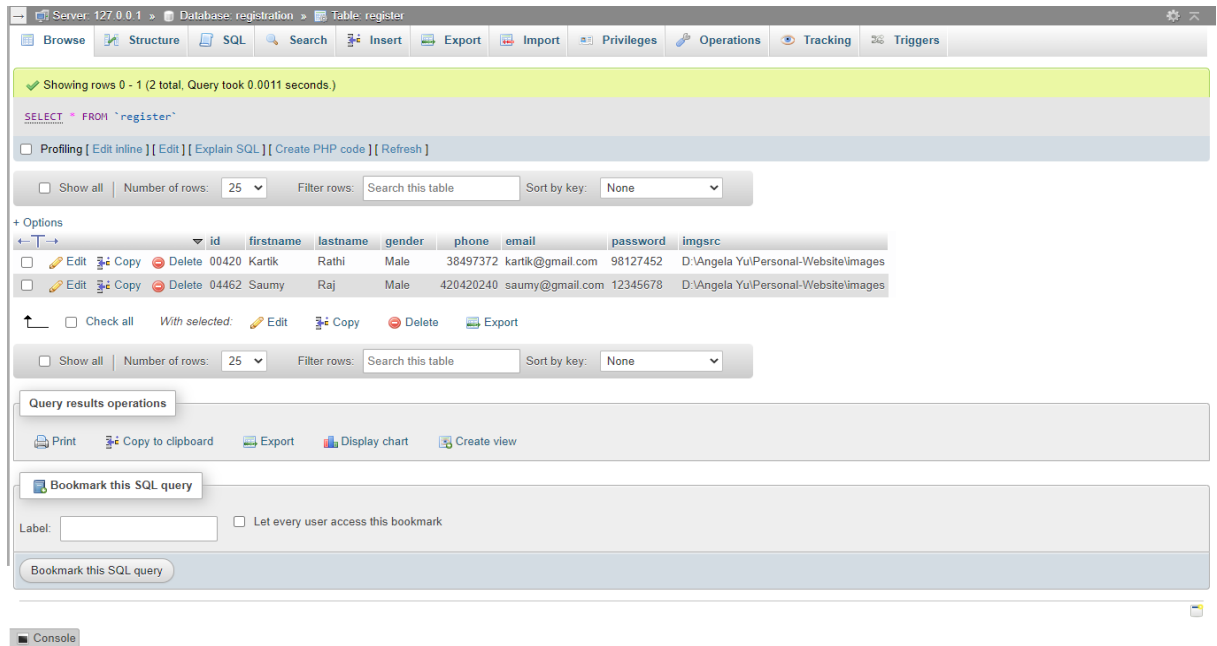


Fig 3.14 – Database for registered students

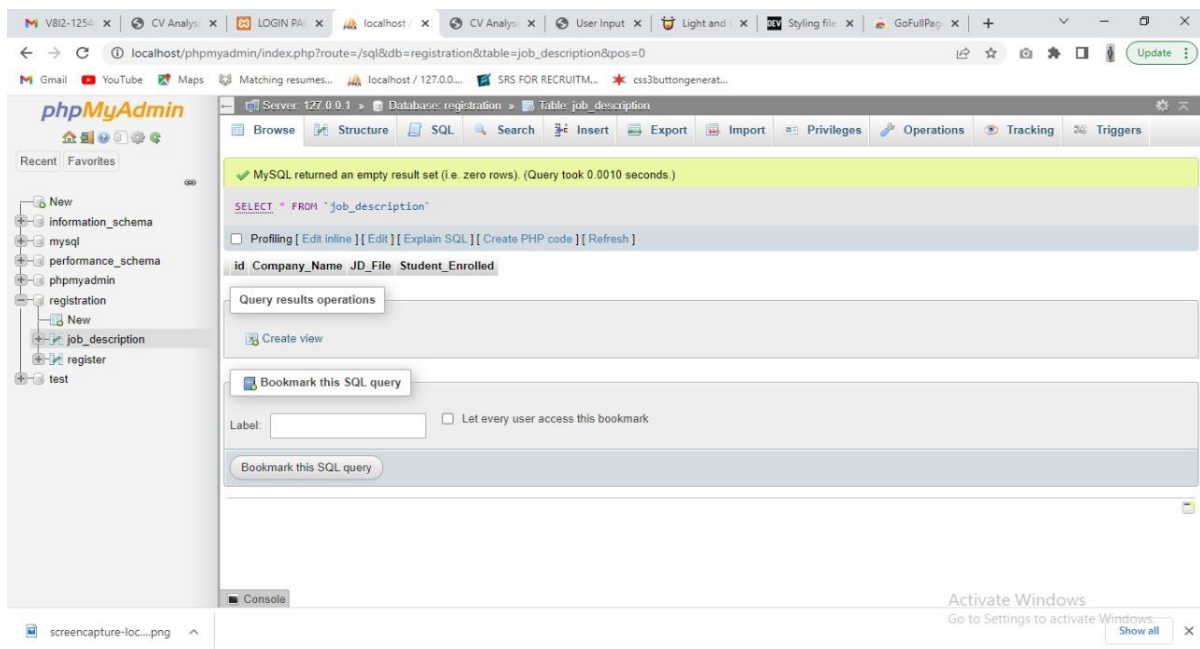


Fig 3.15 – Database for JD's and students enrolled in them

3.3.2 SDLC model for the proposed system

- As the proposed project has an implementation of a machine learning model, so its performance and accuracy are not fixed and it depends on the dataset and other factors so it's best suited for Agile Methodology.
- So if any requirement gets changed or any new dataset gives the best result we can implement that and also we can easily manage the changes. Also, our project has been divided designing in separate phases for each member like - Frontend, Backend, Machine Learning Model (it's testing and training) and it will be merged together after completion of each phase.

3.3.3 Algorithm

1. The HR login in the system using username and password.
2. The system will match the credentials of the user if it doesn't match with the database then he/she needs to sign-up.
3. If the credentials match then he will upload the CV of the candidate in the system.
4. The model will process the CV and give the summarization.
5. Then it matches it with the job description and gives the similarity score in percentages and stores the result in the database.

CHAPTER – 4

IMPLEMENTATION AND DATASET

4.1 System modules and flow of implementations

4.1.1 Loading spaCy model

- You can download spaCy model using `python -m spacy en_core_web_lg`
- Then load spacy model into nlp.

```
In [8]: nlp = spacy.load("en_core_web_sm")
skill_pattern_path = r"C:\Users\ss\AppData\Local\Programs\Python\Python39\Scripts\Major-project(new)\jobzilla_ai-main\jz_skill_pa
```

Fig 4.1 - Code snippet for loading spaCy model

4.1.2 Entity Ruler

To create an entity ruler we need to add a pipeline and then load the .jsonl file containing skills into ruler. As you can see we have successfully added a new pipeline entity_ruler. In our scenario, the entity ruler allows us to apply extra rules to emphasize other categories inside the text, such as talents and job description.

```
In [9]: ruler = nlp.create_pipe('entity_ruler')
ruler.from_disk(skill_pattern_path)
nlp.pipe_names

Out[9]: ['tok2vec', 'tagger', 'parser', 'attribute_ruler', 'lemmatizer', 'ner']
```

Fig 4.2 - Code snippet for Entity Ruler

4.1.3 Skills

We will create two python functions to extract all the skills within a resume and create an array containing all the skills. Later we are going to apply this function to our dataset and create a new feature called skill. This will aid us in seeing the dataset's trends and patterns.

- get_skills is a function which will extract skills from a single text.
- unique_skills will remove duplicates.

```
In [10]: def get_skills(text):  
    doc = nlp(text)  
    myset = []  
    subset = []  
    for ent in doc.ents:  
        if ent.label_ == "SKILL":  
            subset.append(ent.text)  
    myset.append(subset)  
    return subset  
  
def unique_skills(x):  
    return list(set(x))
```

Fig 4.3 - Code snippet for skills

4.1.4 Visualization

Now that we have everything we want, we are going to visualize Job distributions and skill distributions.

4.1.5 Jobs Distribution

The 200 random dataset taken contain various job categories. Accountants, Business development, and Advocates are the top categories.

```
In [13]: fig = px.histogram(
          data, x="Category", title="Distribution of Jobs Categories"
        ).update_xaxes(categoryorder="total descending")
fig.show()
```

Distribution of Jobs Categories

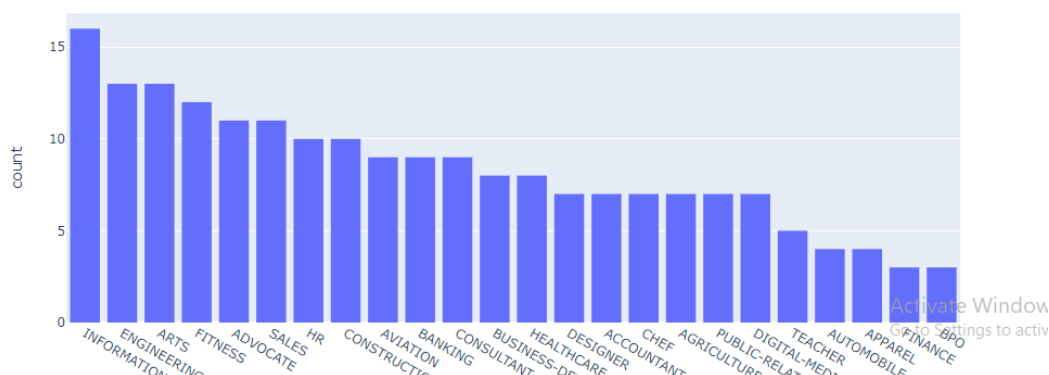


Fig 4.4 - Code snippet for Job description

4.1.6 Entity Recognition

We can also display various entities within our raw text by using spaCy displacy.render. This function is an amazing way to look at your entire document and discover SKILL or GEP within your Resume.

```
In [ ]: sent = nlp(data["Resume_str"].iloc[2])
displacy.render(sent, style="ent", jupyter=True)
```

Fig 4.5 - Code snippet for Entity Recognition

4.1.7 Custom Entity Recognition

In our case, we have added a new entity called SKILL and is displayed in gray color. Apart from changing the color we wanted to add another entity called Job Description so we started experimenting with various parameters within `displace`.

- Adding Job-Category into entity ruler.
- Adding custom colors to all categories.

- Adding gradient colors to SKILL and Job-Category

```
In [ ]: patterns = df.Category.unique()
for a in patterns:
    ruler.add_patterns([{"label": "Job-Category", "pattern": a}])

In [ ]: # options=[{"ents": "Job-Category", "colors": "#ff3232"}, {"ents": "SKILL", "colors": "#56c426"}]
colors = {
    "Job-Category": "linear-gradient(90deg, #aa9cfc, #fc9ce7)",
    "SKILL": "linear-gradient(90deg, #9BE15D, #00E3AE)",
    "ORG": "#ffd966",
    "PERSON": "#e06666",
    "GPE": "#9fc5e8",
    "DATE": "#c27ba0",
    "ORDINAL": "#674ea7",
    "PRODUCT": "#f9cb9c",
}
options = {
    "ents": [
        "Job-Category",
        "SKILL",
        "ORG",
        "PERSON",
        "GPE",
        "DATE",
        "ORDINAL",
        "PRODUCT",
    ],
    "colors": colors,
}
sent = nlp(data["Resume_str"].iloc[5])
disply.render(sent, style="ent", jupyter=True, options=options)
```

Fig 4.6 - Code snippet for Custom Entity Recognition

4.1.8 Resume Analysis

In this part, we are allowing users to copy & paste their resumes and see the results. As we can see after adding the Resume the results are amazing. The model has successfully highlighted all the skills.

```
In [77]: input_resume=tx
sent2 = nlp(input_resume)
disply.render(sent2, style="ent", jupyter=True, options=options)
```

c:\users\kartik\appdata\local\programs\python\python38\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: ``should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during the transform in `preprocessing_exc_tuple` in IPython 7.17 and above.`

Michael Smith PERSON BI / Big Data/ ORG Azure Manchester, UK- Email me on Indeed: indeed.com/r/falicent/140749dace5dc26f 10+ years DATE

of Experience in Designing, Development, Administration, Analysis GPE, Management in the Business Intelligence Data warehousing, Client Server Technologies, Web-based Applications, cloud solutions and Databases PRODUCT. Data warehouse: Data analysis, star/ snow flake schema data modeling and design specific to data warehousing and business intelligence environment. Database PERSON: Experience in database designing, scalability, back-up and recovery, writing and optimizing SQL ORG code and Stored Procedures PERSON, creating functions, views, triggers and indexes. Cloud platform: Worked on Microsoft Azure ORG cloud services like Document DB PERSON, SQL ORG Azure, Stream Analytics ORG, Event hub, Power BI ORG, Web Job, Web App, Power BI PRODUCT, Azure data lake analytics (U-SQL). Big Data ORG: Worked Azure data lake store/analytics for big data processing and Azure data factory to schedule U-SQL ORG jobs. Designed and developed end to end big data solution for data insights. Willing to relocate: Anywhere WORK EXPERIENCE Software Engineer Microsoft - Manchester ORG, UK GPE, December 2015 DATE to Present

Fig 4.7 - Code snippet for Resume Analysis

4.1.9 Match Score

In this section, we are allowing recruiters to add skills and get a percentage of match skills. This can help them filter out hundreds of Resumes with just one button.

```
In [79]: input_resume=tx
req_skills = (input_skills.lower()).split(",")
req_skills = input_skills.split(",")
print(req_skills[0])
resume_skills = unique_skills(get_skills(input_resume.lower()))
score = 0
for x in req_skills:
    if x in resume_skills:
        score += 1
req_skills_len = len(req_skills)
match = round(score / req_skills_len * 100, 1)

print(f"The current Resume is {match}% matched to your requirements")
```

Fig 4.8 - Code snippet for Matching Score

4.2 Dataset

A set of 2400+ resume examples culled from livecareer.com that can be used to sort a resume into any of the classifications indicated in the dataset: Resume Dataset.

4.2.1 Inside the CSV

- ID: Unique identifier and file name for the respective pdf.
- Resume_str: Contains the resume text only in string format.
- Resume_html: Contains the resume data in html format as present while web scrapping.
- Category: Category of the job the resume was used to apply.

Present categories

HR, Designer, Information-Technology, Teacher, Advocate, Business-Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital-Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public-Relations, Banking, Arts, Aviation

4.2.2 Acknowledgements

Data was obtained by scrapping individual resume examples from www.livecareer.com website.

4.2.3 Jobzilla skill patterns

The jobzilla skill dataset is jsonl file containing different skills that can be used to create spaCy entity_ruler. The data set contains label and pattern-> different words used to describe skills in various resume.

4.2.4 Resume Dataset

Using Pandas read_csv to read dataset containing text data about Resume.

- We are going to randomized Job categories so that 200 samples contain various job categories instead of one.
- We are going to limit our number of samples to 200 as processing 2400+ takes time.

```
In [5]: df = pd.read_csv(r"C:\Users\ss\AppData\Local\Programs\Python\Python39\Scripts\Major-project(new)\Resume\Resume.csv")
df = df.reindex(np.random.permutation(df.index))
data = df.copy().iloc[
    0:200,
]
```

Fig 4.9 - Code snippet for loading Dataset

Chapter 5

RESULTS AND TESTING

5.1 Result

We utilized an entity ruler to build more entities in this project, which we subsequently presented utilizing custom colors. We also showed categories and skill distributions, as well as allowing users to easily input resumes with skill match percentages.

We had never utilized spaCy in depth before, so it was a learning process for us. We've also uncovered a number of ways in which our project may be utilized to help enhance the recruiting process by weeding out the best candidates for the position.

5.2 Testing

When we're working with language, measuring the results of our model outputs becomes a lot more complicated. For many NLP-based challenges, this becomes abundantly evident very soon. Now comes the question that how to measure the accuracy of a language-based sequence when it comes to language summarization or translation?

The best answer to this question would be Recall-Oriented Understudy for Gisting Evaluation (ROUGE).

It works by comparing a summary or translation generated automatically with a set of reference summaries (typically human-produced).

- System Summary (what the machine produced)
- Reference Summary (gold standard — usually by humans).

5.2.1 Type of testing adapted

ROUGE-N

ROUGE-N measures the number of matching ‘n-grams’ between our model-generated text and a ‘reference’. An n-gram is simply a grouping of tokens/words. A unigram (1-gram) would consist of a single word. A bigram (2-gram) consists of two consecutive words:

Original: "the girl is standing next door"

Unigrams: ['the', 'girl', 'is', 'standing', 'next', 'door']

Bigrams: [' the girl', 'girl is', ' is standing', ' standing next', 'next door']

Trigrams: [' the girl is', 'girl is standing', ' is standing next', 'standing next door']

The N in ROUGE-N stands for the n-gram that we're employing. The match-rate of unigrams between our model output and the reference would be measured for ROUGE-1. Bigrams and trigrams would be used by ROUGE-2 and ROUGE-3, respectively. We now need to determine whether we want to compute the ROUGE recall, precision, or F1 score once we've decided which N to employ.

Recall

The recall divides the total number of n-grams in the reference by the number of overlapping n-grams identified in both the model output and the reference.

$$\text{Recall} = (\text{number of n-grams found in model and reference}) / (\text{number of n-grams in reference})$$

This is helpful for verifying that our model captures all of the information in the reference, but it isn't so great for ensuring that our model isn't simply pumping out a lot of vocabulary to manipulate the recall score.

Precision

To circumvent the recall issue, we utilize the precision metric, which is generated in a similar manner but divides by the model n-gram count rather than the reference n-gram count.

$$\text{Precision} = (\text{number of n-grams found in model and reference}) / (\text{number of n-grams in model})$$

F1-Score

Using recall and precision we can calculate our ROUGE F1 score like so:

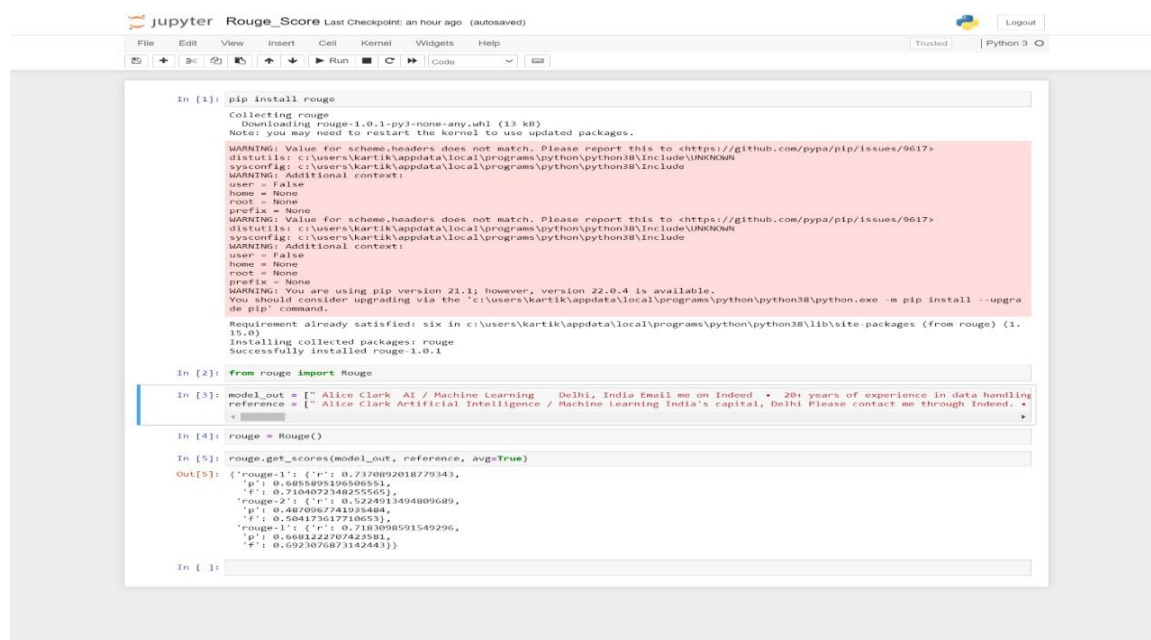
$$\text{F-1 Score} = 2 * (\text{precision} * \text{recall} / \text{precision} + \text{recall})$$

This provides a trustworthy measure of our model's performance, as it relies not only on the model catching as many words as possible (recall), but also on it not outputting unnecessary words (precision).

5.2.2 Test result of various stages

Fortunately, implementing these metrics in Python is incredibly easy thanks to the Python rouge library. We can install the library using the command- “pip install rouge”.

And scoring our model output against a reference is as easy as this:



```

In [1]: pip install rouge
Collecting rouge
  Downloading rouge-1.0.1-py3-none-any.whl (13 kB)
Note: you may need to restart the kernel to use updated packages.
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/9617>
distutils: c:\Users\kartik\appdata\local\programs\python\python38\include\UNKNOWN
sysconfig: c:\Users\kartik\appdata\local\programs\python\python38\include
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/9617>
distutils: c:\Users\kartik\appdata\local\programs\python\python38\include\UNKNOWN
sysconfig: c:\Users\kartik\appdata\local\programs\python\python38\include
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
WARNING: You are using pip version 21.1; however, version 22.0.4 is available.
You should consider upgrading via the 'c:\Users\kartik\appdata\local\programs\python\python38\python.exe -m pip install --upgr
de pip' command.
Requirement already satisfied: six in c:\Users\kartik\appdata\local\programs\python\python38\lib\site-packages (from rouge) (1.
15.0)
Installing collected packages: rouge
Successfully installed rouge-1.0.1

In [2]: from rouge import Rouge

In [3]: model_out = [" Alice Clark AI / Machine Learning  Delhi, India Email me on Indeed  20+ years of experience in data handling
reference = [" Alice Clark Artificial Intelligence / Machine Learning India's capital, Delhi Please contact me through Indeed.  "]

In [4]: rouge = Rouge()

In [5]: rouge.get_scores(model_out, reference, avg=True)
Out[5]: {'rouge-1': {'r': 0.7370092018779343,
  'p': 0.6858095196506531,
  'f': 0.7104072348255565},
 'rouge-2': {'r': 0.5224913494809689,
  'p': 0.4870967741935484,
  'f': 0.504173617718053},
 'rouge-l': {'r': 0.7183098591549296,
  'p': 0.6681222707423581,
  'f': 0.6923076873142443}}

In [ ]:

```

Fig 5.1 - Code snippet for Rouge Score

The `get_scores` method returns three metrics, ROUGE-N using a unigram (ROUGE-1) and a bigram (ROUGE-2) — and ROUGE-L.

For each of these, we receive the F1 score f , precision p , and recall r .

Typically we would be calculating these metrics for a set of predictions and references — to do this we format our predictions and references into a list of predictions and references respectively — then we add the `avg=True` argument to `get_scores`.

5.2.3 Conclusion of testing

Rouge-1 “recall” - 73.70%. This means that 73% words in the reference summary have been captured by the system summary.

Rouge-1 “precision” – 68.55%. This simply means that 68% words in the system summary were in fact relevant or needed.

Rouge-1 “F1 score” – 71.04%.

Rouge-2 “recall” – 52.24%. It tells that the system summary has recovered 52% bigrams out of total bigrams from the reference summary.

Rouge-2 “precision” – 48.70%. The precision here tells us that out of all the system summary bigrams, there is a 48% overlap with the reference summary.

“F1 score” – 50.41%.

As the summaries (both system and reference summaries) get longer and longer, there will be fewer overlapping bigrams. This is especially true in the case of abstractive summarization, where you are not directly re-using sentences for summarization. The reason one would use ROUGE-1 over or in conjunction with ROUGE-2 (or other finer granularity ROUGE measures), is to also show the fluency of the summaries or translation. The intuition is that if you more closely follow the word orderings of the reference summary, then your summary is actually more fluent.

CHAPTER 6

CONCLUSION AND FUTURE IMPROVEMENT

6.1 Conclusion

The field of text summarizing is such exciting experimental topic among NLP community which helps to give short and accurate information in a clear and clean manner. The main purpose of this paper is to provide the latest study, progress, and research which is made in this field till date. From all the above models it is concluded that with the use of the NER method, we can provide a systematic, comprehensive, clear, and wide review, from the field of styles/topics, data sets, different coping strategies, problems, available assessment methods as a guide to future work. NER works best on the group of resumes as it works for extractive summarization so, it summarizes the given resume according to the words present in the resume without altering or changing them which makes it more precise for HR's point of view to see the summarized document.

Extractive summaries still hold the top of current popular trend topics in this research, even though they are far simpler than the most complicated abstractive summaries, which are quite complex. This is due to the fact that more study is required and that many questions remain unanswered in the abstractive summarization process, which is a hurdle that researchers must overcome. It can also be shown that semantics, similarity, sentence position, sentence length, frequency, keywords, and the necessity to be there are the most essential variables in making a good or clean summary.

6.2 Future Improvement

Future work in this field of textual summary research could include: i) solving problems related to feature, such as picking features to employ in data summarization to discover the more

appropriate features, uncovering new features, creating the most often utilized features, using a variety of semantic features, finding the best factors to produce coherent sentences, and adding system elements. ii) Preprocessing the database problem with the right title; otherwise, POS Tagging is necessary to prevent word deletion and create tokens, and this is done to distinguish word categories such as nouns, adjectives, verbs, and so on. iii) Summing up the mathematical methodologies, machine learning, and fuzzy-based is the most difficult to try in the extractive summaries. iv) We can enhance the current methodologies, such as NATSUM in some circumstances, or increase NATSUM performance by boosting compliance, by using abstractive summaries. v) Unusual datasets, such as legal papers, tourism attractions summaries, and inspection documents summaries. Alternatively, we might utilize prominent online data sets like DUC 2002 and 2004 to assess the summarizing process before examining private data to ensure the method's accuracy.

REFERENCES

- [1] Linet, Nikitha. (2020). “ATS BREAKER”- A System for Comparing Candidate Resume and Company Requirements. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS060403.

- [2] Roy, P.K., Chowdhary, S.S., & Bhatia, R. (2020). A Machine Learning approach for automation of Resume Recommendation system. Procedia Computer Science, 167, 2318-2327.

- [3] Yadav, Vikas & Bethard, Steven. (2019). A Survey on Recent Advances in Named Entity Recognition from Deep Learning models.

- [4] Lin, Yiou & Lei, Hang & Addo, Prince & Li, Xiaoyu. (2016). Machine Learned Resume-Job Matching Solution.

- [5] Schmitt, T., Caillou, P., & Sebag, M. (2016). Matching Jobs and Resumes: a Deep Collaborative Filtering Task. GCAI.

- [6] Kudatarkar, V.R., Ramannavar, M.M., & Sidnal, N.S. (2015). An Unstructured Text Analytics Approach for Qualitative Evaluation of Resumes.

- [7] Liu, Y., & Lapata, M. (2019). Text Summarization With Pretrained Encoders. Arxiv, Abs/1908.08345

- [8] Narayan, Shashi & Cohen, Shay & Lapata, Mirella. (2018). Ranking Sentences for Extractive Summarization with Reinforcement Learning.

- [9] Zhang, Xingxing & Lapata, Mirella & Wei, Furu & Zhou, Ming. (2018). Neural Latent Extractive Document Summarization.

- [10] Liu, Y., Titov, I., & Lapata, M. (2019). Single Document Summarization as Tree Induction. NAACL.

- [11] Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural Document Summarization by Jointly Learning to Score and Select Sentences. *ACL*.
- [12] See, A., Liu, P.J., & Manning, C.D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *ACL*.
- [13] Asli, Celikyilmaz & Bosselut, Antoine & He, Xiaodong & Choi, Yejin. (2018). Deep Communicating Agents for Abstractive Summarization.
- [14] Paulus, Romain & Xiong, Caiming & Socher, Richard. (2017). A Deep Reinforced Model for Abstractive Summarization.
- [15] Narayan, Shashi & Cohen, Shay & Lapata, Mirella. (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization.
- [16] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *ACL*.
- [17] Honnibal, M. and Montani, I., 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.
- [18] Akbik, A., Blythe, D. and Vollgraf, R., 2018, August. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1638-1649).
- [19] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 265-283).
- [20] Narayan, Shashi & Cohen, Shay & Lapata, Mirella. (2018). Ranking Sentences for Extractive Summarization with Reinforcement Learning.
- [21] Nallapati, Ramesh & Zhou, Bowen & Dos Santos, Cicero & Gulcehre, Caglar & Xiang, Bing. (2016). Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. 280-290. 10.18653/v1/K16-1028.
- [22] Chung, J., Gülcehre, Ç. Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv*, abs/1412.3555.
- [23] Gibadullin, Ilshat & Valeev, Aidar. (2020). Experiments with LVT and FRE for Transformer model.
- [24] Mok, Peggy. (2010). Language-specific realizations of syllable structure and vowel-to-vowel coarticulation. *The Journal of the Acoustical Society of America*. 128. 1346-56. 10.1121/1.3466859.
- [25] Nomoto, Tadashi. (2005). Bayesian Learning in Text Summarization. 10.3115/1220575.1220607.

- [26] El-Kassas, Wafaa & Salama, Cherif & Rafea, Ahmed & Mohamed, Hoda. (2020). Automatic Text Summarization: A Comprehensive Survey. *Expert Systems with Applications*. 165. 113679. 10.1016/j.eswa.2020.113679.
- [27] Ozsoy, Makbule & Alpaslan, Ferda & Cicekli, Ilyas. (2011). Text summarization using Latent Semantic Analysis. *J. Information Science*. 37. 405-417. 10.1177/0165551511408848.
- [28] Steinberger, Josef & Jezek, Karel. (1970). Text Summarization and Singular Value Decomposition. 3261. 245-254. 10.1007/978-3-540-30198-1_25.
- [29] Gong, Yihong & Liu, Xin. (2001). Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*. 19-25. 10.1145/383952.383955.
- [30] Steinberger, Josef & Jezek, Karel. (2004). Using Latent Semantic Analysis in Text Summarization and Summary Evaluation.
- [31] Murray, Gabriel & Renals, Steve & Carletta, Jean. (2005). Extractive summarization of meeting recordings. 593-596. 10.21437/Interspeech.2005-59.
- [32] Linhares Pontes, Elvys & Huet, Stéphane & Torres-Moreno, Juan-Manuel & Linhares, Andréa. (2018). Cross-Language Text Summarization Using Sentence and Multi-Sentence Compression. 10.1007/978-3-319-91947-8_48.
- [33] Nagwani, Naresh. (2015). Summarizing large text collection using topic modeling and clustering based on MapReduce framework. *Journal of Big Data*. 2. 10.1186/s40537-015-0020-5.
- [34] Kutlu, Mucahid & Cigir, Celal & Cicekli, Ilyas. (2010). Generic Text Summarization for Turkish. *Comput. J.* 53. 1315-1323. 10.1093/comjnl/bxp124.
- [35] Guo, Q., Huang, J., Xiong, N.N., & Wang, P. (2019). MS-Pointer Network: Abstractive Text Summary Based on Multi-Head Self-Attention. *IEEE Access*, 7, 138603-138613.
- [36] Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç. Xiang, B., 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*. pp. 280–290. <https://doi.org/10.18653/v1/k16-1028>.
- [37] Nallapati, R., Zhai, F., Zhou, B., 2017. SummaRuNNer: A recurrent neural network-based sequence model for extractive summarization of documents, in: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. pp. 3075–3081.
- [38] Subali, Made & Fatichah, Chastine. (2019). Kombinasi Metode Rule-Based dan N-Gram Stemming untuk Mengenali Stemmer Bahasa Bali. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 6. 219. 10.25126/jtiik.2019621105.

- [39] Fábio Bif Goularte, Silvia Modesto Nassar, Renato Fileto, Horacio Saggion, A text summarization method based on fuzzy rules and applicable to automated assessment, *Expert Systems with Applications*, Volume 115, 2019 - <https://doi.org/10.1016/j.eswa.2018.07.047>.
- [40] Foong, Oi-Mean & Oxley, Alan. (2011). A hybrid PSO model in Extractive Text Summarizer. *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*. 10.1109/ISCI.2011.5958897.
- [41] Müller, Š. & Gaikwad, D.K. (2020). Text Summarization Using Named Entity Recognition.
- [42] Khan, Rahim & Qian, Yurong & Naeem, Sajid. (2019). Extractive based Text Summarization Using KMeans and TF-IDF. *International Journal of Information Engineering and Electronic Business*. 11. 33-44. 10.5815/ijieeb.2019.03.05.
- [43] Azmi, Aqil & Altmami, Nouf. (2018). An abstractive Arabic text summarizer with user controlled granularity. *Information Processing and Management*. 54. 903-921. 10.1016/j.ipm.2018.06.002.
- [44] Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Syst. Appl.*, 68, 93-105.
- [45] A. Alzuhair and M. Al-Dhelaan, "An Approach for Combining Multiple Weighting Schemes and Ranking Methods in Graph-Based Multi-Document Summarization," in *IEEE Access*, vol. 7, pp. 120375-120386, 2019, doi: 10.1109/ACCESS.2019.2936832.
- [46] Robertson, S. (2004), "Understanding inverse document frequency: on theoretical arguments for IDF", *Journal of Documentation*, Vol. 60 No. 5, pp. 503-520. <https://doi.org/10.1108/00220410410560582>