Automatic resume segmentation and screening the resume using NLP and machine Learning methods

Prabal Kumar Chowdhury
Dept. of CSE
BRAC University
prabal.kumar.chowdhury@g.bracu.ac.bd

Muhtasim Mahmud
Dept. of CSE
BRAC University
muhtasim.mahmud@g.bracu.ac.bd

Mohammad Rahat Khan
Dept. of CSE
BRAC University
md.rahat.khan@g.bracu.ac.bd

***Abstract*—There are now a great number of firms beginning their path, especially software enterprises. In addition, there are other well-established enterprises on the market. All of these businesses need workers with distinct skill sets and educational backgrounds. Therefore, it is exceedingly difficult for all of these large firms to manually examine and manage all of the resumes of applicants. There is already a substantial amount of work in NLP that can automatically partition the resume to handle this issue. Consequently, these study works on automated CV segmentation merely make the lives of recruiters simpler; what about candidates? In an effort to simplify the lives of both recruiters and applicants, we have suggested a model that can automatically segment the candidate's curriculum vitae, as well as filter the candidate's curriculum vitae and make recommendations based on previously approved resumes for the position. Thus, each applicant may modify his or her resume in accordance with the recommendations and get their ideal job. The model provided is based on NLP and ML.**

***Index Terms*—NLP methods, Machine Learning, topic modeling, Natural Language Processing, text segmentation, word embedding**

Introduction

In large organisations, where the number of applicants is large and CVs must be processed manually, this information extraction and parsing issue is very real and poses a significant drain on time and manpower. In this setting, the duty of automated CV parsing to extract information about the candidate's already accepted for the position includes information about their work experience, education, skills, further education, and general information. We are extracting information about the talents that helped them get the position by analysing their resumes. As long as the retrieved data allows for faster resume processing, it might be utilised to lessen the burden on the recruiter. This saves time and effort for the recruiter who previously had to make the final choice among applicants. He need just trust on their prior choices, as the NLP technique will handle the rest. Now let's look at the other side of the coin: the applicants. The job-seeking student has no idea why his CV was rejected or what he can do to make it more competitive in the future. Our methodology starts by analysing each applicant's resume using natural language processing techniques, after which it is compared to resumes that were successful in landing the advertised position. Then, using machine learning, we can estimate how likely it is that this resume will be accepted. Our approach will provide feedback to applicants whose resumes were not chosen for further consideration, advising them on how to rework their applications to increase their chances of being hired.

Problem Statement

Particularly in large organisations with a high volume of new applicants, at which manual CV computation is inefficient and taxing on scarce human resources, automatic CV (or resume) information extraction and parsing is an indispensable process. Automatic CV parsing becomes an issue here because it must be done in order to extract data such as the candidate's work history, education, skills, and so on. However, many students who apply for the role have no idea why his resume was rejected or how he can strengthen and update it to become more competitive.

Research Objectives

This research's main aim is to build an automatic resume segmentation and screening model. By using the combination of NLP and machine learning method we will create this model which can help both the recruiters and job candidates also. The objectives are given below :

- Making recruiters life easier by creating this automatic screening model
- Help the candidates to find their percentage to get any specific job role
- Suggest candidates how to improve their resume by related skills for their desired job role.

Literature Review

[1] When filling a position in an organisation, Pradeep Kumar Roy developed a system that uses a candidate's resume to perform natural language processing (NLP), named entity recognition (NER), and text classification (using machine learning and n-gram) for a 39% success rate, a 44% success rate using multinomial naive bayes (MNB), a 64% success rate using logistic regression (LR), and a 71% success rate using linear support vector machine (LSVM).
[2] Thimma Reddy Kalva also did some data analysis work using a web service API for her study. Comparison between candidates were performed for the suitable position evaluating their qualifications. I used NER tools (such as Apache OpenNLP and Stanford Name Entity Recognizer) to complete

the task.
[3] Yong Luo headed a team of researchers that studied private resume management firm data, the vast majority of which was unlabeled but which included some tagged information. The study categorised the data into two groups which are positive and negative.
[**?**] Sujit Amin used two hundred resumes in his study of a web tool for screening applications. Clients, servers, and personnel finder personnel all used the programme. After a candidate submits their resume, the server will rate the resumes and send the results to the recruiter's end.
[5] Tejaswini K suggested a process in which the applicant would submit their resume after the completion of an MCQ exam that was equipped with a face recognition technology to identify instances of cheating. It was accomplished via the use of NLP strategies and the application of TF-IDF vectorization in order to ensure that the machine understood it. By using KNN algorithm as the classifier the closest to the job description criteria was determined which was provided by the candidates and the accuracy of the result was 83 %.
[**?**] A team including Riza Tanaz Fareed, Rajath V, and Sharadadevi Kaganumath developed and implemented resume using cosine similarity. The resume goes through a natural language processing pipeline where the keywords are pulled out. To find the right vocabulary, specialists use strategies like stop words and lemmatization. For the KNN model to properly categorise the resume, the words are vectorized with the help of the TF-IDF vectorizer. This trained model has an accuracy of 98.96%.
[7] Suhas Tangadle Gopalakrishna and Vijayaraghavan Varadharajan conducted a study by using semantic analysis which is used to explain their procedure. To remove filler words like "and," "or," "the," and "etc." from the resumes we receive, we run them through NLPP and employ the Stop deletion feature. There are also supplementary methods used, such as NER and Parts of Speech tagging. Six distinct classification models are employed, including Naive Bayes, Multinomial Naive Bayes, Linear Support Vector Machines, Bernoulli Naive Bayeses, Logistic Regression, and K-Nearest Neighbors. Multinomial Nave Bayes had the highest accuracy (91%) among the six classifiers.
[8] By using NER, Cosine similarity, NLP Suhas H E and Manjunath A E tried to develop a model which proposes job roles for the candidates. Dump of technical abilities used for skill-tagging resumes. Tab The produced separated value TSV0 file is then used to train the Stanford NER model. The shallow neural network word2vec model takes its input from the NER model. Cosine similarity was used to compare resumes, and it was able to find a match between 79.8% of them.

DATASET

In this case, we're comparing two different cv formats. To be more specific, the first kind is the standard. Therefore, we are sorting through the resumes that have been approved. Two specialised recruiters were contacted, and from them, 200 acceptable resumes were received. The second kind of resumes is the standard form used by applicants. Therefore, we sourced them from regular college seniors who are applying to various businesses.

METHODOLOGY

The final model that we have suggested is, in essence, a fusion of two different models. The Natural Language Processing (NLP) approach comes first, followed by the Machine Learning (ML) method.
The proposed method conducts three class classifications (basic information, education, and job experience) for each line, allowing a CV to be broken down into segments where neighbouring lines are grouped together based on their shared characteristics. In the first stage of CV prepossessing and

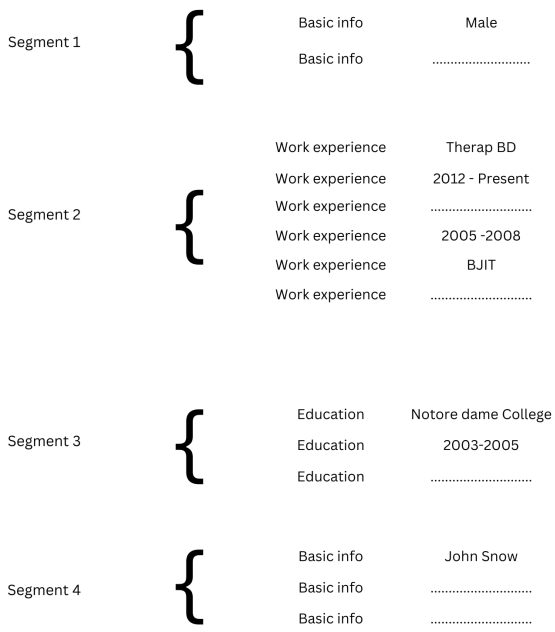

Fig. 1. Overall process

transformation, resumes are parsed line by line from pdf (or docx) files. The text is then converted into a form that can be read and understood by machine learning algorithms. In particular, the algorithm iteratively scans the text in a sliding window that moves down by one line at a time, ultimately selecting a portion of text that

- begins on the current line;
- consists of at least 25 words;
- Where it ends in the text corresponds to the end of a line.

Thus, we acquire a set of text windows for each CV that match exactly to the set of text lines in the CV.
Constructing Word Embeddings The next stage involves inserting the CVs collection's words into the linear space Rn. For each word, we therefore have a vector with n dimensions. As an added bonus, many word embedding methods are available,

each of which builds word embeddings that retains syntactic and semantic aspects of the words they work with. Word2Vec, FastText, and GloVe are three of the more well-known options. Furthermore, many multiple linguistic have pre-trained word embeddings that are constructed from vast collections of texts and may be used without cost.
Tf-Idf Index Calculation The third stage involves calculating the tf-idf index for each word in each text window. The tf-idf statistic may be used to gauge a word's relative significance within a given manuscript. Word frequency in individual papers is negatively related to overall document frequency. For this reason, tf-idf can identify uncommon terms that are overrepresented in a given text. Because of this, it is often employed as a ranking criteria in the field of information retrieval.
Document d from the collection has the tf-idf analysis performed on it for the term t. tf idf(t, d, D) = tf(t, d) idf(t, d, D) is the formula used to get the value of D. (t, d, D). (3) D. Building Embedded Text Fields in Text After obtaining the CV components, the n-dimensional vector is embedded with each component. The words in the CV are represented as vectors, and the tf-idf weights of those words are added together for this purpose. Word w in document d, CV text field t, v(d), v(w) — embeddings for document d and word w in document d, tf- idf(w, d) — tf- idf index of word w in document d. This means that each text window is converted into an n-dimensional feature vector that might be fed into a machine learning algorithm.
Extraction of Unique Characteristics To improve classification outcomes in the long run, more features are added on top of that increased feature area.

- Counting Part-of-Speech — the number of each grammatical category shown in the text box. Eleven brand new characteristics, each of which corresponds to a distinct component of speech, are added in total.
- The amount of words in the text box that have suffixes that are unique to the description of the applicant's professional experience and education. In the version of the method for suffix characteristics that is currently in use.

This data has been correctly segmented, and we are currently saving it in a csv file. As we complete initial screening of all incoming resumes, we save each line from the segmented resumes in a csv file, along with the relevant information such as skills, education, and work experience. Therefore, we will use this csv dataset to train our ML model to predict, for every given job position, what proportion of randomly generated fresh resumes will be approved. A job description and skill set are read from the csv file and used to train the model. So, when a student inputs their résumé and desired position, the system can predict whether or not they will be hired for that function. If our ML model determines that you have a low likelihood of being hired for this position, it will use natural language processing to provide suggestions about other positions for which you could be a better fit or about the ways in which your CV relates to the posting.

Experimental Result Analysis

We will use data we scraped off of Kaggle's open platform to train the model. The first model, K-Nearest Neighbor or Support Vector Machine, predicts the type of job for which a given resume is best suited. Based on this prediction, the second model suggests ways to strengthen the resume through the use of cosine similarity, which is determined by comparing the user's desired job role with the model's prediction.
In this field there are many work already exist which are in the reference given bellow. So here we are comparing our model with the other models also :

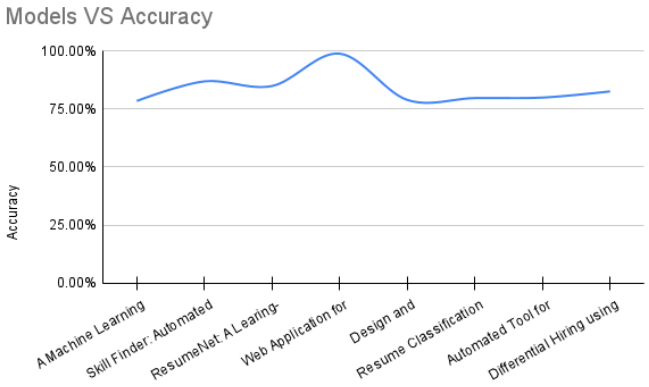

Fig. 2. Overall process

In our model, there are also some more processes. That is to say, when we take resumes from students or applicants, we apply the NLP approach to their resumes using the exact same procedure as before. When we have their resumes in hand, we segment them once again before moving on to the next phase. At this point, we have completed two more steps in comparison to the initial NLP technique. Classification and complete segmentation are the two remaining tasks to complete.

Table I: Model parameters

Parameter name	Work Experience	Education
n_estimators (catboost)	500	500
learning_rate (catboost)	0.1	0.09
depth (catboost)	9	8
l2_leaf_reg (catboost)	10	10
smoothing window size	10	6
gap-window size	5	12

Fig. 3. Model Parameters

So after receiving the resume and expected job position, we are first segment the resume and then classifies the resume. Then

we extract the specific segments/features from the resume and give them to our ML method. Then from the previous dataset of accepted cv's, our ML method do the screening and all. So here for the extra two steps of NLP methods have to work fully again and here we are showing the model parameters.
On the test set, the Jaccard index was separately calculated for job experience and education. The outcomes using the aforementioned settings were as follows: Jaccardeducation = 0.806 and Jaccardwork experience = 0.942. Additionally, the quantity of CVs with clear segments was counted. Definition. When Jwork exp(C) + Jedu(C) $>$ 1.7, a CV is said to be well-segmented.

CONCLUSION

In the article, a novel approach to automatically parsing and segmenting CVs was discussed. This approach makes it possible to glean information on a candidate's previous employment and educational background from a text document that is either in pdf or docx format. In the near future, one of our goals is to improve the quality of the algorithm by enhancing the training data with nonstandard samples of CVs. In addition, we want to develop the algorithm further in order to enable the extraction of new segments, such as talents, more education, and so on.
This paper discusses a number of techniques that may be used to detect, identify, and categorise different resumes. These techniques make use of a number of different machine learning and Neural Network models, such as SVM, KNN, Word2Vec, and Cosine similarity, among others. The accuracy of the models may vary anywhere from 78% to 98%, depending on the datasets that are utilised, the complexity of the learning techniques, and the quantity of the dataset. The results can be anywhere from those two extremes. In conclusion, we have demonstrated that it is possible to attain the level of accuracy and output that is desired for a wide range of applications by utilizing the appropriate dataset in conjunction with the appropriate method. This was done by showing that it is possible to achieve the level of accuracy and output that was sought.
We are always working to increase the accuracy of our model, despite the fact that it already has a reasonable degree of precision. Our primary objective is to create a web application that employs our model on the backend in order to make the results of our study accessible to the average recruiter and applicant. If we are successful in this endeavour, it will be beneficial to both the applicant and the recruiter. The candidate will submit their resumes through this web application, and if the resume is likely to accept only then that resume will be forwarded to the recruiter. If, on the other hand, the candidate does not possess the appropriate skills for this job, then our model will suggest some skills and works in which the candidate should focus to get this job.

REFERENCES

[1] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space."arXiv preprint arXiv:1301.3781 (2013).

[2] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations."Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013.

[3] Bojanowski, Piotr, et al. "Enriching word vectors with subword information."Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

[4] Joulin, Armand, et al. "Bag of tricks for efficient text classification."arXiv preprint arXiv:1607.01759 (2016)

[5] Pradeep Kumar Roy, Vellore Institute of Technology, 2019. A Machine learning approach for automation of resume recommendation system, ICCIDS 2019. 10.1016/j.procs.2020.03.284.

[6] Thimma Reddy Kalva, Utah State University, 2013. Skill-Finder: Automated Job-Resume Matching system. 3]Yong Luo, Nanyang Technological University, 2018. A LearningBased Framework for automatic resume quality assessment, arXiv:1810.02832v1 cs.IR].

[7] Suhjit Amin, Fr.Conceicao Rodrigues Institute of Technology, 2019. Web Application for Screening resume, IEEE DOI: 10.1109/ICNTE44896.2019.8945869.

[8] Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna, Design and Development of Machine Learning based Resume Ranking System (2021), DOI: https://doi.org/10.1016/j.gltp.2021.10.002

[9] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation."Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[10] Y. Luo, Y. Wen, T. Liu, and D. Tao, "Transferring knowledge fragments for learning distance metric from a heterogeneous domain," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.