LJAST
Libyan Journal of Applied Science and Technology
مجلة ليبيا للعلوم التطبيقية والتقنية

Volume 11 Issue 02
December 2023
ISSN 2958-6119

Analysis of Resumes and Recommendations for Job Matches

Esam Elsheh[1], Husam Elgomati[2]
[1]School of Advanced Technology, Algonquin College, Ottawa, Canada
[2]Information Technology Dept., College of Engineering Technology–Janzour, Tripoli, Libya
elsh0049@algonquinlive.com[1], hus_7119@yahoo.com[2]
Received 19 August 2023; revised 25 August 2023; accepted 25 August 2023

الملخص

مهمة البحث عن مرشحين مناسبين لوظيفة شاغرة يمكن أن تكون مرهقة، خاصةً عند وجود عدد كبير من المتقدمين. يمكن أن يعيق هذا فريق الموارد البشرية في العثورعلى الشخص المناسب في الوقت المناسب لتلك الوظيفة. ومع ذلك، يمكن أن يخفف بشكل كبير من هذه العملية العبئة عن طريق أتمتة عملية تصنيف السير الذاتية ومطابقتها بواسطة توفيرعملية فحص واختصار عادلة. يمكن لهذا النظام التعامل مع عدد كبير من السير الذاتية أولاً من خلال تصنيفها في فئات مناسبة باستخدام مصنفات متنوعة. بمجرد اكتمال عملية التصنيف، يتم مقارنة السيرة الذاتية المدخلة مع السير الذاتية الأخرى في الفئة باستخدام التشابه Cosine Similarity لاسترجاع أفضل المرشحين الذين سيتم تصنيفهم استنادًا إلى وصف الوظيفة.

يستخدم هذا النظام التشابه Cosine Similarity والتصويت الأغلبي مع (Logistic Regression, Naïve Bayes, SVM, Decision Tree, and Random Forest) لتصنيف السير الذاتية والعثور على أقرب تطابق لوصف الوظيفة المقدم، مما يؤدي إلى عملية اختيار مرشحين أكثر كفاءة وعملية اتخاذ القرار.

Abstract

The task of finding suitable candidates for an open role can be overwhelming, particularly when there are numerous applicants. This can hinder the team's progress in finding the right person within a reasonable timeframe. However, automating the process of resume classification and category matching can greatly alleviate this burdensome process by providing fair screening and shortlisting. In this paper, we propose an automatic resume recommender to match the posted job description. This system can handle a large number of resumes by first classifying them into appropriate categories using various classifiers. Once classification is complete, the input resume then is compared with the other resumes in the category using Cosine similarity to retrieve the top candidates which will be ranked based on the job description.

This system utilizes cosine similarity and Majority voting with (Logistic Regression, Naïve Bayes, SVM, Decision Tree, and Random Forest) to classify the resume and find the closest match to the provided job description, resulting in a more efficient candidate selection and decision-making process.

Keywords: Resume Recommender, Natural Language Processing, Machine Learning.

1. Introduction

The current job market presents a challenging environment for job seekers who struggle to distinguish themselves from other applicants. This paper seeks to address this issue by developing a Resume Recommender system that utilizes natural language processing techniques. The system will assist employers in identifying potential resumes that align with their job requirements. The research will utilize Python programming language and a range of NLP libraries to achieve its objectives. Several projects and research papers proposed similar resume recommendations based on different techniques. The technique proposed in [1] used KNN and cosine similarity, and in [2] the author vectorized the resumes and computed the matchings using Cosine similarity. The K-means++ was

used in [3] for clustering for job positions and combines TF-IDF and part-of-speech weights to improve job matching, resulting in improved recommendations over word frequency vectors. In [4], the paper employed various metrics, including cosine similarity, soft cosine similarity, Jaccard similarity, dice similarity coefficient, overlap coefficient, and conditional probability. The research concluded that conditional probability outperforms the other metrics, making it the selected metric for similarity assessment. In [5], they introduced an automatic system, which enhances fair screening, accelerates candidate selection through category classification, content-based recommendations, utilizing cosine similarity, and k-NN for closest job description matches.

2. Dataset

The dataset was acquired from Kaggle.com under the name Resume Dataset published by Snehaan Bhawal. This is a dataset of 2483 resumes in PDF format, gathered from Kaggle. The dataset includes examples of resumes categorized into specific labels. Each resume is stored in the "data" folder and is organized by label, with each label having its own folder. The resumes are in PDF format, with the file name being the ID of the resume. This dataset can be used to classify a given resume into one of the predefined labels.

2.1. Resumes Categories

The dataset contains resumes from 24 different categories as follows, 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, Figure 1. Shows the distribution of the topic categories.

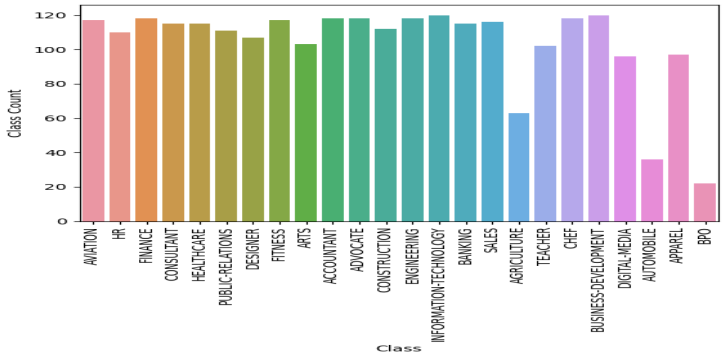

Figure 1. Categories Distribution

3. Preprocessing

We imported all the PDF resume files into a Pandas DataFrame. Within the DataFrame, we labeled each resume according to the category folder it belongs to.

	file_name	Text	Class
0	21287405.pdf	P AVIATION SUPPLY SPECIALIST Summary Ambitious...	AVIATION
1	11698189.pdf	HR EMPLOYEE RELATIONS SPECIALIST Summary Dedic...	HR
2	21912637.pdf	CONTRACTS FINANCE OFFICER Professional Profile...	FINANCE
3	27726066.pdf	CONSULTANT Summary HR Professional nearly year...	CONSULTANT
4	24548333.pdf	SENIOR SPECIALTY SALES REPRESENTATIVE Summary ...	HEALTHCARE
...	...	...	...
2479	10820510.pdf	QA QC MANAGER Summary QA QC Manager Qualificat...	CONSTRUCTION
2480	19938081.pdf	CUSTOMER CARE REPRESENTATIVE Professional Summ...	FITNESS
2481	96493528.pdf	V P COMMERCIAL RELATIONSHIP MANAGER Summary Ac...	BANKING
2482	18937778.pdf	DIRECTOR GLOBAL BUSINESS DEVELOPMENT Summary S...	BUSINESS-DEVELOPMENT
2483	31948488.pdf	FINANCE MANAGER Summary Highly organized detai...	FINANCE

Figure 2. Resumes Dataframe

3.1 Data Cleaning

Data cleaning is an important step in preparing text data for natural language processing tasks. It involves removing noise, irrelevant information, and inconsistencies in the text data to improve the accuracy and reliability of the analysis.

a. Remove non-letters Characters

We use the regular expressions to remove any non-letter characters

```
import re
regex = re.compile('[^a-zA-Z]')
resumes_df['Text'] = resumes_df['Text'].apply(lambda x: regex.sub(' ', x))
```

b. Removing stop-words

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
resumes_df['Text'] = resumes_df['Text'].apply(remove_stopwords)
```

Volume 11 Issue 02
December 2023
ISSN 2958-6119

c. Remove punctuation

```
import string
punct_pattern = r'[{}]'.format(string.punctuation)
resumes_df['Text'] = resumes_df['Text'].str.replace(punct_pattern, '')
```

d. Convert to lower case

```
resumes_df = resumes_df.applymap(lambda x: x.lower() if type(x) == str else x)
```

e. Lemmatization

This involves reducing words to their base or root form. For example, "running" and "ran" would both be reduced to "run". This can help to reduce the number of unique words in the dataset and simplify the analysis.

```
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```


3.2 Vocabulary Size

After performing all the above steps the total vocabulary size was 34187, these words were derived from the 24 different resumes categories.

4. Resume Classification

Resume classification is a task in natural language processing (NLP) that involves automatically assigning a resume to one or more categories based on its content. The categories could be job titles, industry sectors, skill sets, or any other relevant classification scheme.

The task of resume classification typically involves several steps, including data preprocessing, feature extraction, model training, and evaluation. In the data preprocessing step, the raw text of the

Volume 11 Issue 02
December 2023
ISSN 2958-6119

resume is cleaned, tokenized, and transformed into a structured format that can be used for analysis. In the feature extraction step, relevant features such as keywords, phrases, and other linguistic patterns are extracted from the preprocessed text.

Once the features have been extracted, they can be used to train a machine learning model to classify resumes into different categories. The model can be trained using supervised learning techniques, where labeled examples of resumes and their corresponding categories are used to train the model.

Once the model has been trained, it can be used to classify new resumes that have not been seen before. The accuracy of the model can be evaluated using a variety of metrics such as precision, recall, and F1 score. If the accuracy of the model is high enough, it can be used to automatically classify resumes into different categories, which can save time and effort for recruiters and hiring managers.

Overall, resume classification is a useful tool for automating the task of resume screening and matching and can help recruiters and hiring managers more efficiently identify candidates that are a good fit for their open positions.

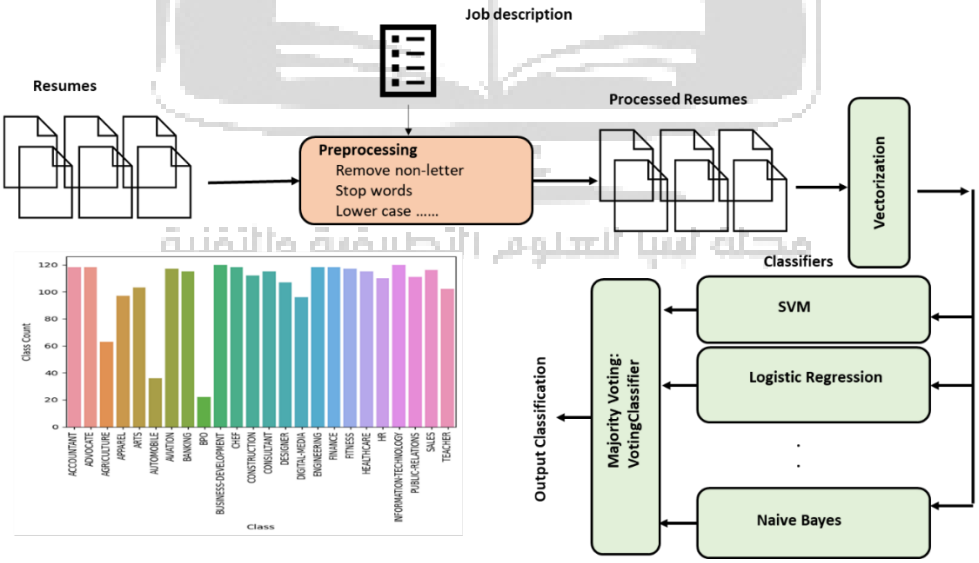

Figure 3. Resume Classification

4.1 Classification

Classification involved employing five distinct models, with accuracy scores carefully documented for each.

Logistic Regression [10]: Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict a binary outcome (e.g. yes/no, true/false, 0/1) based on one or more input variables or features. In logistic regression, the output variable is modeled as a probability of belonging to a particular class, using a logistic function or sigmoid function.

Naïve Bayes [7]: Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem, which describes the probability of an event occurring based on prior knowledge of conditions that might be related to the event.

Support Vector Machines [8]: Support Vector Machines (SVMs) are a popular machine learning algorithm used for classification tasks. SVMs attempt to find a decision boundary that separates the different classes in the input data with the largest possible margin.

Random Forest [6]: Random Forest is a popular machine learning algorithm used for classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to improve performance and reduce overfitting.

Decision Tree [9]: Decision Tree is a machine learning algorithm used for classification. It works by constructing a tree-like model of decisions and their possible consequences based on the input data.
In this paper we used the algorithms mentioned above for the resume classification. The Table 2. shows the metric measures of each algorithm.

Table 1. Classification Algorithms measures

Classifier	Accuracy	F1 Score	Recall	precision
Logistic Regression	68.49	64.73	64.47	66.42
Naïve Bayes	38.87	36.54	36.04	45.31

Support Vector Machines	59.24	53.46	53.85	57.27
Random Forest	67.61	57.80	59.45	64.39
Decision Tree	58.84	54.47	55.22	55.07

4.2 Majority Voting

The majority vote is a technique used to combine the predictions of multiple classifiers in order to make a final prediction. The idea is to have each classifier in the ensemble independently make a prediction, and then have the ensemble choose the most commonly predicted class as the final prediction.

The majority vote technique is particularly useful when individual classifiers have high accuracy on average, but can occasionally make incorrect predictions. By combining the predictions of multiple classifiers, the ensemble can achieve higher accuracy and be more robust to noisy data.

```
lr = LogisticRegression(max_iter=300)
svm = SVC(probability=True,  random_state = 0)
tree = DecisionTreeClassifier(max_leaf_nodes=200, random_state=0)
nb = GaussianNB()
rfc = RandomForestClassifier(n_estimators=200, random_state = 0)

ensemble = VotingClassifier(estimators=[('lr', lr), ('svm', svm),('tree', tree), ('rfc', rfc)],
voting='hard')
ensemble.fit(X_train, y_train)
y_pred = ensemble.predict(X_input_resume)
```

5. Resume Category Matcher

A resume category matcher is a tool that can be used to check if a given resume matches a desired job category or not. This tool can be useful for both job seekers and employers, as it can help job seekers

tailor their resumes to a specific job category, and it can help employers quickly identify resumes that are relevant to their open positions.

The basic idea behind a resume category matcher is to use natural language processing (NLP) techniques to analyze the text of a resume and compare it to a set of job categories or descriptions. The NLP techniques can be used to identify keywords, phrases, and other patterns in the text that are associated with a particular job category.

For example, suppose a company is looking to hire a software engineer. The company can create a job description that includes keywords and phrases such as "software development", "coding", "Java", "Python", "database design", and so on. The company can then use a resume category matcher tool to scan through resumes and identify those that contain similar keywords and phrases.

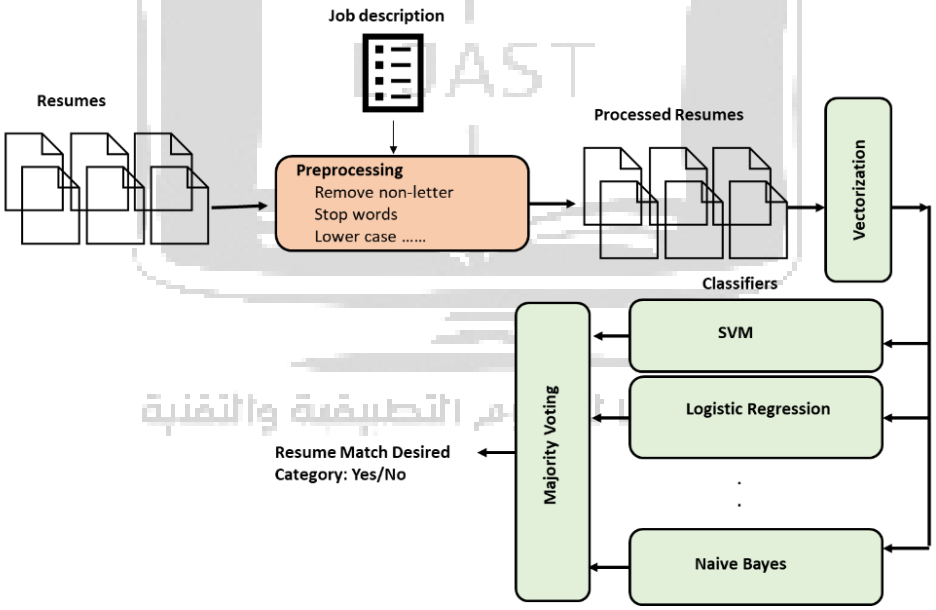

Figure 4. Resume Category Matching

6. Finding Similar Resumes

Finding the resume similarity from a pool of resumes to the job description is a task in natural language processing (NLP) that involves measuring how closely each resume in the pool matches the job description. The goal is to identify the resumes that are most similar to the job description and therefore most likely to be a good fit for the position.

Volume 11 Issue 02
December 2023
ISSN 2958-6119

The first step in this task is to preprocess the job description and the resumes in the pool. This involves cleaning the text, removing stop words, and stemming the words to reduce the number of variations in the text.

Next, relevant features such as keywords, phrases, and other linguistic patterns are extracted from the job description and the resumes. These features can be used to represent the text as numerical vectors that can be compared using a variety of similarity metrics such as cosine similarity, Jaccard similarity, and Euclidean distance.

Once the features have been extracted, a similarity score can be calculated for each resume in the pool. The score indicates how closely the resume matches the job description, with higher scores indicating a closer match.

Finally, the resumes can be ranked based on their similarity scores, with the highest ranking resumes being the ones that are most similar to the job description. The recruiter or hiring manager can then review the top-ranked resumes to determine which candidates to invite for further interviews or consideration.

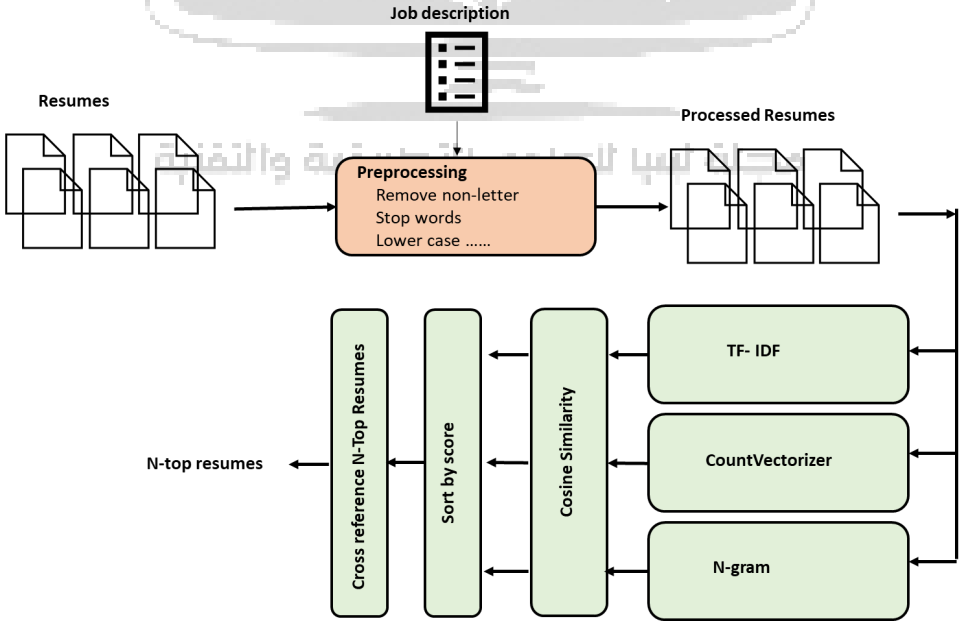

Figure 5. Resume Recommendation based on NLP Similarity

7. NLP fundamental methodologies

We have applied the following fundamental methodologies in the domain of natural language processing and text analysis. These techniques facilitate the conversion of textual data into numerical representations and the computation of text similarity, which plays a crucial role in a variety of NLP and machine learning applications.

a. Vectorization

Vectorization in natural language processing (NLP) refers to the process of converting text into numerical vectors that can be used as input to machine learning models. This is an important step in NLP because most machine learning algorithms require numerical data as input, whereas text data is inherently non-numerical.

We vectorized the resumes and the input job description using three methods as follows,

b. TF-IDF vectorization

This technique involves creating a vector for each word in the text, where the vector contains a weighted score that reflects how important the word is in the text relative to other texts.

c. CountVectorizer vectorization

This technique involves creating a vector for each word in the text, where the vector contains the count of the word in the text.

d. n-gram CountVectorizer vectorization

The ngram_range parameter in CountVectorizer specifies the range of n-grams to be included as features in the vectorized output. For example, CountVectorizer(ngram_range=(1, 3)) will include unigrams (single words), bigrams (two-word sequences), and trigrams (three-word sequences) as features in the vectorized output.

Once the text has been vectorized, the resulting vectors can be used as input to machine learning models. These models can then be trained to perform a variety of NLP tasks such as text classification, sentiment analysis, named entity recognition, and machine translation.

e. Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors in a multi-dimensional space. In natural language processing (NLP), cosine similarity is commonly used to determine the similarity between two documents or pieces of text.

The cosine similarity between two vectors is calculated as the cosine of the angle between them. The value ranges from -1 to 1, with higher values indicating greater similarity. A cosine similarity of 1 indicates that the vectors are identical, while a cosine similarity of 0 indicates that the vectors are completely dissimilar.

	file_name	Text	Class	CosSimilarity_TFIDF	CosSimilarity_CVec	CosSimilarity_ngram	count
17	83338413.pdf	field support specilist summary technology sup...	arts	0.077608	0.134538	0.069269	3
4	12011623.pdf	engineering quality technician career overview...	engineering	0.075519	0.164997	0.090938	3
8	18297650.pdf	volunteer hr ivolunteer summary sponsorship re...	hr	0.055892	0.128820	0.071350	2
16	55595908.pdf	site engineering career overview year total in...	engineering	0.043384	0.118430	0.063097	2
15	43378989.pdf	consultant summary depth knowledge understandi...	consultant	0.058467	0.136753	0.075926	2

Figure 6. Top 5 Matching Resumes

8. Conclusion

In this paper, we explored three different machine learning techniques to identify the most suitable resume that matches a given job description. Specifically, we utilized Resume Classification, Resume Category Matching, and Resume Similarity systems, each of which can work independently based on the employer's requirements. Our analysis was based on the dataset that containing 2483 resumes and 24 topic categories.

To achieve better accuracy in classification and similarity analysis, we recommend using a much larger dataset that covers a wider range of job categories. For instance, datasets such as the Open Resume Dataset, which contains over 3 million resumes, or the JobTech Resume Dataset, which includes 200,000 resumes, may provide more comprehensive insights and higher accuracy in the classification and matching of resumes with job descriptions.

9.References

[1] P. Roy, S. Chowdhary and R. Bhatia "A Machine Learning approach for automation of Resume Recommendation system" ICCIDS 2019.

[2] J. Wang "How to Build a Resume Recommender like the Applicant Tracking System (ATS)" Towards Data Science, 2020.

[3] L. Duan, X. Gui, M. Wei, Y. Wu "A Resume Recommendation Algorithm Based on K-means++ and Part-of-speech TF-IDF", AIAM 2019 International Conference on Artificial Intelligence and Advanced Manufacturing, 2019.

[4] S. M. Shovon, M. Bin Mohsin, K. Jahan Tama, J. Ferdaous and S. Momen "An Automated CV Recommender System Using Machine Learning Techniques", Data Science and Algorithms in Systems. CoMeSySo 2022. LNCS, vol 597. Springer, 2023.

[5] P. Kumar Roy, S. Singh Chowdhary, R. Bhatia "A Machine Learning approach for automation of Resume Recommendation system", Procedia Computer Science, vol. 167, pp. 2318-2327, 2020.

[6] L. Breiman, Random forests. Machine learning 45, pp.5–32, 2001.

[7] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, pp. 41–46, 2001.

[8] B. Scholkopf, A.J. Smola, F. Bach, et al., Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[9] N.E.I Karabadji, H. Seridi, F. Bousetouane, W. Dhifli, S. Aridhi, S., An evolutionary scheme for decision tree construction. Knowl.-Based Syst. 119, pp. 166–177, 2017.

[10] A.F. Cabrera, Logistic regression analysis in higher education: An applied perspective. Higher Education: Handbook of Theory and Research, vol. 10, pp.225–256, 1994.