Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230937

Resume recommendation based on text similarity

Zhenyu Dong

School of Software Engineering, Tongji University, Shanghai, 201800, China

1852143@tongji.edu.cn

Abstract. With the vigorous development of the mobile Internet era, traditional offline recruitment methods are gradually fading out of people's sight. Internet recruitment has become the first choice of many job seekers and companies due to its advantages of low cost and high efficiency. However, suitable positions for job seekers and resume information for recruiters are often drowned in a sea of information. How to find an effective resume recommendation algorithm is still an open issue. To this end, this paper proposes a personalized resume recommendation method based on text classification. Specifically, by encoding and extracting the feature information of different resumes, and scoring the matching degree between job seekers and job information based on the XGBoost algorithm, personalized job information recommendations are made for job seekers. A large number of experiments have verified the effectiveness of the method in this paper, which can provide new insights for the efficient matching of resumes and job information.

Keywords: Resume Classification, Resume Recommendation, XGBoost.

1. Introduction

With the vigorous development of the mobile Internet era, traditional offline recruitment methods are gradually fading out of people's sight, replaced by Internet recruitment with a huge amount of information. Job hunting based on the Internet has the advantages of wide coverage, timely, low cost, and high efficiency, which has become the first choice of many job seekers and companies. According to the "2021 China Online Recruitment Industry Development Report", there will be 5.267 million online recruiting employers in 2021, an increase of 3.5% over the previous year [1]. It is estimated that in 2022, the overall number of online recruitment employers will exceed 6 million. However, the problem of information overload also arises, where the positions suitable for job seekers and resume information of recruiters are often submerged in massive information. In this context, in order to achieve mutual matching between a resume and job information, it is imminent to find an effective resume recommendation algorithm.

A resume is a comprehensive display of a person and many years of experience concentrated, whose diversity, authenticity, and uniqueness can not be replaced by other personal material. However, in most cases, the use of a resume is the initial screening before the interview, which has a lot of valuable information that is not explored. The wealth of information in a resume creates a variety of dimensions that can be explored. The resume contains the basic dimensions of school and major, one's gender and age, the mastery of one's skills, and the project of internship work experience. Different industries have different requirements for different positions, so it is a prerequisite for job

DOI: 10.54254/2755-2721/6/20230937

recommendation to determine how well these dimensions match those of the job seekers who are working or have worked in the field.

A resume recommendation system is essentially an information flow product. Different from recommendation systems widely used in e-commerce, the resume recommendation system is relatively simple due to its single demand, where the user profile of job seekers need not be as thousands or even tens of thousands of tags in the user profile of e-commerce [2]. The first resume recommendation system was proposed by Bradley et al. in 2000, which is based on case retrieval methods and collaborative filtering. Since then, Cambazoglu formulated the resume recommendation task for current employees as a supervised machine learning problem by mining the historical job change information of job-seeking employees. Hong et al. divided users into different clusters and adopts different recommendation methods for different user clusters. The system can intelligently select recommendation methods according to the characteristics of users. Shivam et al. proposed a content-based recommendation engine with similarity techniques using N-Grams and topic model approaches. Yang et al. build a true hybrid job recommendation system by combining statistical relational learning (SRL) with content-based and model-based recommendation algorithms.

Based on summarizing and analyzing the content of previous research, this paper proposes a personalized resume recommendation method based on text classification. Specifically, we code and extract the characteristic information of different resumes and score the matching degree between job seekers and job information based on the XGBoost algorithm, and make personalized job information recommendations for job seekers.

The assignment of this paper is as follows. Section 1 is the introduction, which mainly describes the research background and the status of existing recommendation systems. In section 2, we analyze the theories related to the topic and introduce our method in detail [3]. We further show the specific experimental procedure and analyze the data of the experimental results in Section 3. Section 4 is the discussion, which summarizes the problems encountered in the task and presents an outlook on future work.

2. Method

In this section, we introduce the whole pipeline of the model construction in detail. We collect 10,000 real and non-duplicate resumes from the Internet, which will be cleaned and structured to get a representative 1500 pieces of data. Then the most appropriate label of each piece of data is provided with a keyword extraction method to provide the most appropriate label for each piece of data. The whole data set processed will be split and tokenized, which will finally be fed to the XGBoost model to classify and predict resumes.

2.1. Data preprocessing

The dataset used in this experiment is from GitHub, which contains about 30,000 pieces of data. This file represents the dataset of resumes in a single text file. Each line of the file contains information about a text resume. Each line has 3 fields separated by ":::". The first field is the reference id of the resume. The second field is the list of occupations separated by ";", and the third field is the text resume. For each data, one resume text may correspond to multiple job labels.

One of the main forms of preprocessing is filtering out useless data. In natural language processing, useless words (data) are called stop words. We can instantly recognize that certain words have more meaning than others. We can also see that some words are useless and filler words [4]. We don't want these words to take up space in our database or take up the valuable processing time. Therefore, we call these words "garbage words" because they have no use and we don't want to deal with them.

To make the data more standardized, this experiment selected 1500 more standardized data from the original large amount of data, which were selected based on whether they contained the complete resume content and the correct job title. These extracted data were then stored in a suitable data structure to facilitate subsequent processing. Meanwhile, in order to simplify the situation that a resume text corresponds to multiple tags, this experiment decided to select one of the many tags of

DOI: 10.54254/2755-2721/6/20230937

each data as its final tag. The method of selecting the tags was to count the frequency of each position appearing in all the data and sort them from highest to lowest to obtain a word frequency table of position tags [5]. Then based on this table, among the labels of each data, the one with the highest word frequency is selected as its final label. In this way, a dataset with a one-to-one correspondence between tags and resume text is obtained.

2.2. Word separation based on NLTK

Performing some form of analysis or processing so that the machine somehow understands what the text means, expresses, or implies is key to natural language processing. However, computers cannot handle text data and need a way to convert words into numbers or signal patterns [6]. NLTK is a Python library commonly used in the field of NLP research, a module developed by Steven Bird and Edward Loper at the University of Pennsylvania based on Python. NLTK is an efficient Python-built platform for processing human natural language data. It provides easy-to-use interfaces through which to access over 50 corpora and lexical resources (such as WordNet), as well as a set of text-processing libraries for classification, tokenization, stemming, parsing, and semantic reasoning, and a wrapper for industrial-grade NLP libraries and an active discussion forum.

In this study, NLTK(Natural Language ToolKit) was used for word separation. Firstly, we divide the resume text into sentences and divide the sentences into words. Then we delete the stop words and punctuation marks in these words, and finally, delete the data with tokens less than 100.

2.3. Boost algorithm

The boost algorithm is an integration algorithm that takes multiple weak classifiers and integrates them to form a strong classifier [7]. Take the decision tree as an example, single decision tree time complexity is low, the model is easy to show, but easy to overfit. The boost method for decision trees is an iterative process in which new training is performed to improve the previous result. The traditional boost method is to weigh the correct and incorrect samples, and after each step, increase the weight of the wrong points, i.e., increase the number of wrong samples, and decrease the weight of the right points, i.e., decrease the number of right samples.

2.4. Model construction based on XGBoost

XGBoost is a gradient-boosted decision tree implementation developed for competitive machine learning speed and performance. XGBoost is a more sophisticated variant of gradient boosting; instead of training models in isolation, boosting trains models in succession, with each new model being trained to rectify the errors committed by the preceding ones. Models are introduced successively until no more advancements are possible. The advantages of XGBoost include: (1) the loss function is approximated by a Taylor spread binomial; (2) the structure of the tree is constrained by regularization to prevent over-complexity of the model and reduce the possibility of over-fitting; (3) the node splitting after optimization derivation. A wrapper class from XGBoost enables models to be used in the scikit-learn framework as classifiers or regressors. As a result, we may employ XGBoost models along with the entire Scikit-Learn library [8].

A word embedding trained in a neural network model on a particular natural language processing job is known as an embedding layer. The task's documents or corpus are cleaned and processed, and the vector space's dimensions—50, 100, or 300—are defined as part of the model. Small random integers are used to initialize the vectors. The Backpropagation technique is used to fit the embedding layer in a supervised manner on the front end of a neural network. It was employed in this work together with GloVe.Since then, it has been the de facto norm for creating pre-trained word embeddings. For ease of usage, we convert the GloVe file containing the word embeddings to the word2vec format [9]. The function glove2word2vec from the gensim.

We must deal with whole sentences, thus we must develop sentence embedding. Basically, this means that every characteristic of the vector will be dependent on the word embeddings. There are numerous choices, but we won't dive into them here. Instead, we use a very straightforward way. In

DOI: 10.54254/2755-2721/6/20230937

order to convert each resume (in this case, a phrase) into a vector representation, we will first construct a class that contains our vocabulary and glove vectors. The last step is to develop a Vectorizer object, which will assist in turning our resumes into vectors, a numerical representation. Then, we can utilize those vectors as inputs for our classifier. XGBoost is a gradient-boosted decision tree implementation that was created for speed and performance in a very competitive machine-learning environment. XGBoost is an improved form of gradient boosting. Instead of training each model independently from the others, boosting trains models sequentially, with each new model being trained to fix the mistakes produced by the prior ones [10].

3. Experiment and performance analysis

After the training of the model, the training results of 53 posts were finally obtained in this experiment. Five of them with a large amount of data were selected, as shown in the following Table 1.

Table 1. Predicted results for most common jobs.

Label	Precision	Recall	f1-score	Support
Database Administrator	0.72293	0.80496	0.76174	282
Front End Developer	0.66000	0.78261	0.71609	253
IT Security Analyst	0.53629	0.61290	0.57204	217
Network Administrator	0.53182	0.49576	0.51316	236
Systems Administrator	0.44510	0.51546	0.47771	291

As shown in Table 1, the model's Precision for Database Administrator is 0.72293 and Recall is 0.80496, and the f1-score obtained after weighing the accuracy and recall rate is 0.76174, which is about 20% higher than the prediction data of IT Security Analyst. This indicates that the resumes of candidates for positions such as Database Administrator and Front End Developer are unique and can be easily determined from their resume text. rate is only 0.51546, indicating that the skills or experiences mentioned in their resume text are closer to those of other job seekers.

After sorting and calculating, the results obtained are shown in the following Table 2：

Table 2. Average predicted results.

	Precision	Recall	f1-score	Support
Accuracy			0.57891	1432
Macro avg	0.07330	0.07342	0.07167	1432
Weighted avg	0.52652	0.57891	0.54883	1432

Some results show that the accuracy rate is low because the amount of data is relatively small, such as the position of Network Engineer, there are 25 pieces of data, but his correct rate is only 42.85%. The accuracy rate is high when there are many of data. Some results are biased because they only take one but there are many labels. There are several data with a large amount of data, but the accuracy is not so high because the data of two labels are highly coincident with each other.

4. Discussion

4.1. Issues

The use of natural language processing in resume analysis has driven the automation of the entire recruitment process. Resume information extraction is technically accomplished by using the dictionary and rule-based, machine learning-based, and deep learning-based methods; text clustering or text classification methods are used to classify resumes; resume recommendation can be achieved by using keyword-based matching, similarity-based and classifier-based methods, depending on the data and requirements [11]. Although the existing NLP-based resume analysis methods have achieved

DOI: 10.54254/2755-2721/6/20230937

a large number of excellent research results and given birth to many brand-new research ideas, there are still some shortcomings and defects that have not been solved.

(1) Lack of resources such as resume corpus. At present, most of the resume data sets for resume analysis technology research are collected by scholars from the Internet and other channels, and there is a lack of professional authoritative, and comprehensive resume text corpus or historical recruitment data for scholars to conduct experimental research and comparative evaluation of the effectiveness of different methods [12].

(2) Poor generality of resume analysis models. Since there is no unified writing standard for resume text, there is great variability in resume text written by different job seekers, which may lead to the model that has completed training on a certain resume dataset and achieved high accuracy being less accurate when tested on other datasets. This puts a high demand on the universality of the analysis model, or it can be solved by unifying the specification of resume format writing.

(3) The accuracy of the resume analysis model needs to be further improved. In recent years, deep learning-based models have improved the accuracy of various tasks in resume analysis, but the improvement is still limited. Scholars are still trying to improve the old methods and combine the newly proposed models to further improve the accuracy of the models.

4.2. Future outlook

According to the current research status, the future application of natural language processing in resume analysis can be researched from the following aspects.

(1) Authoritative academic and research institutions or assessment benchmark websites should develop resume annotation specifications, and annotate and publish resume corpus resources for scholars to conduct resume analysis research [14].

(2) Training on multiple data sets. To improve the generality of the analysis model, resume data sets from different regions, industries and languages can be used for joint training to enhance the generality and robustness of the model.

(3) Application of attention mechanism. In recent years, attention mechanism has become one of the research hotspots in natural language processing, but the research on attention mechanism in resume analysis is still relatively small, and there will be more and more research results on resume analysis based on attention mechanism in the future.

(4) Design of end-to-end model. Integrate resume information extraction, resume classification, and resume recommendation together, and build an end-to-end resume analysis model for joint training to achieve more accurate resume recommendations.

(5) Introduction of question and answer task. The question-and-answer task in natural language processing is applied to resume analysis, for example, the job description requirements can be converted into questions and the answers into a list of candidates.

5. Conclusion

In contemporary corporate recruitment, recruitment experts need to rely not only on existing knowledge to judge information such as job seekers' education and school, but also to consider whether the work experience of job seekers' resumes meets the job requirements and descriptions. In this paper, a model is established to classify resumes concerning the recruitment characteristics of Internet companies. The main research includes investigating and studying the current mainstream resume recommendation algorithms, including resume recommendation algorithms based on personalized recommendation, resume recommendation algorithms based on deep learning, and resume recommendation algorithms based on ontology domain knowledge. By analyzing and summarizing the characteristics of the current main algorithms, a model applicable to resume text classification is established. In addition, how to extract more complete, abstract, and high-level semantic features to achieve a textual semantic representation of resumes and recruitment.

DOI: 10.54254/2755-2721/6/20230937

References

[1] Li S.W., Shu F., Guang Y., Zhai Y., Yang Z.J.. A review of research on the application of natural language processing in resume analysis[J]. Computer Science,2022,49(S1):66-73.

[2] Yu, Legang. Design and System Implementation of IT Job Search Resume Scoring Model[D]. Shandong University, 2022. DOI:10.27272/d.cnki.gshdu.2022.001864.

[3] Shi Yuanpeng,Shan Jianfeng. Research on resume matching recommendation algorithm based on text similarity[J]. Computer Simulation,2022,39(04):441-444+491.

[4] Li J, Zheng Y, Tang YH, Wang JC. Data analysis of college students' resumes based on machine learning[J]. Science and Technology Innovation and Application,2022,12(02):83-86.DOI:10.19981/j.CN23-1581/G3.2022.02.021.

[5] Shi, Yuanpeng. Research on two-way matching recommendation algorithm of resume and job information based on text similarity[D]. Nanjing University of Posts and Telecommunications, 2021. DOI:10.27251/d.cnki.gnjdc.2021.000480.

[6] Pradeep Kumar Roy,Sarabjeet Singh Chowdhary,Rocky Bhatia. A Machine Learning approach for automation of Resume Recommendation system[J]. Procedia Computer Science,2020,167(C).

[7] Zhao Yongsheng. Research on job search recommendation system for college graduates[D]. Jiangsu University of Science and Technology, 2020. DOI:10.27171/d.cnki.ghdcc.2020.000256.

[8] Hao K. Design and implementation of job recommendation system based on resume data[D]. Southeast University,2018.

[9] Guo Jie. Research on personalized resume recommendation algorithm for IT industry[D]. Northwestern University,2018.

[10] Research on personalized resume recommendation algorithm for online recruitment [C]//. Proceedings of the Twenty-fifth Chinese Academic Conference on Databases (I). ,2008:87-90.

[11] Zuo Yuqian. Research on human post matching method based on BiLSTM and XGBoost [D]. Dalian University of Technology, 2022. DOI:10.26991/d.cnki.gdllu.2022.000473.

[12] Song, W.T.. Research on hybrid recommendation algorithm based on XGBoost and SVD [D]. Shenyang Normal University, 2022. doi:10.27328/d.cnki.gshsc.2022.000071.

[13] Wang Zhaoli. Design and research of job matching system in online recruitment [D]. Shanghai Jiaotong University,2015.DOI:10.27307/d.cnki.gsjtu.2015.000309.

[14] Liu Lina. Research on E-Recruiting platform for property industry based on personalized information service[D]. Beijing Forestry University,2014.

[15] Song Qingqing. Ontology-based research on user modeling of personalized recommendation system for online recruitment [D]. Nanjing University of Aeronautics and Astronautics,2009.