Automatic Extraction of Segments from Resumes using Machine Learning

Gunaseelan B
Digital Technology Solutions
Mirafra Software Technologies Pvt Ltd
Bangalore, India
gunaseelanb@mirafra.com

Supriya Mandal
Digital Technology Solutions
Mirafra Software Technologies Pvt Ltd
Bangalore, India
supriyamandal@mirafra.com

Rajagopalan V
Digital Technology Solutions
Mirafra Software Technologies Pvt Ltd
Bangalore, India
raja@mirafra.com

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342596

Abstract— **Online recruitment systems or automatic resume processing systems are becoming more popular because it saves time for both employers and job seekers. Manually processing these resumes and fitting to several job specifications is a difficult task. Due to the increased amount of data, it is a big challenge to effectively analyze each resume based on various parameters like experience, skill set, etc. Processing, extracting information and reviewing these applications automatically would save time and money. Automatic data extraction, focused primarily on skillset, experience and education from a resume. So, it extremely helpful to map the appropriate resume for the right job description. In this research study, we propose a system that uses multi-level classification techniques to automatically extract detailed segment information like skillset, experience and education from resume based on specific parameters. We have achieved state-of-the-art accuracy in the segment of the resumes to identify skill sets.**

Keywords—Information Retrieval, IR, resume parsing, segmentation, classification, bagging, boosting, GBDT and random forest.

I. Introduction

Online or intelligent recruitment systems play a significant role in the recruitment channel with the advanced text processing techniques. In recent years, most of the companies have posted their job requirements on various online platforms or gather a large number of resumes from different sources. Recruitment industry spends an excessive amount of time and energy on filtering and pulling data for job requirements from resumes. The entire resume review process based on a job description is quite tedious because the resume is in unstructured format. Therefore, in recent years, many recruitment tools have been developed for the retrieval of information. Although fundamental theories and processing methods for web data extraction are exist, most of the recruitment tools are still subject to text processing and candidates complying with job requirements. Chif et al.[1] use a technique of ontology matching to distinguish technical skills with a set of common multi-word part-of-speech patterns, such as lexicalized, multi-word language. If the specific skill is not mentioned in the database, then it fails to recognize the information to extract from resume. Named-entity recognition (NER) has been widely studied in NLP to identify and extract name, entities from text. With the increased computational power, deep learning models like LSTM and RNN also being used for named-entity recognition. M. Zhao et al. [2] proposed NER and named entity normalization (NEN), to extract relevant information from resumes. NER refers to the identification of a phrase that expresses skill, while NEN goes further and relates the phrase to a specific individual or entity. The authors define a SKILL-considered application which detects, extracts and links a skill to a qualified target. Most of the recruitment systems invest more time to structure the information from unstructured resume by several techniques. The skill tagging matches the skills found in resumes with the concepts of skill taxonomy, which needs to be developed for each domain. The automation of this task developed by using MediaWiki and rules established on category tags. Chen et al. [3] integrated the text block classification by defining the text as a separate blocks, with the help of title words and writing style of the document in resumes. But it will be more complicated for the unstructured resume because the writing style and the title words vary in resumes. There are some NLP techniques to understand the context in the text. Hence, the heuristic model has been used to retrieve the information from resume based on lexical, syntactic and semantic analysis [4]. Unlike classification model, Bernabé-Moreno et al. [5] approached the word embedding technique, with Glove model to mine the skill information by capturing the semantic relationships, between the terms and also clustered the entities with the help of t-SNE. Sequential models like LSTM and RNN are extensively used for sequence tagging task in text, speech, and images. Ayishathahira et al. [6] developed a Convolutional Neural Network (CNN) segmentation model that segmented the knowledge into the family, educational, occupational, and others. In their research, CRF and Bi-LSTM-CNN created for information extraction. They have also shown CRF outperforms the other neural networks, based on the segmentation accuracy. Kumar, R et al. [7] proposed a supervised learning approach, by deep learning technique to predict keyword relevance to identify the skills from resume. Hence, they took skills as relevant keywords and others as irrelevant keywords in that approach. Then they measured the importance of each word in the text, which reflects the probability of a word is the skill in the document. Feature extraction and feature selection are important steps for the text classification, to understand the pattern and reduce number of features from the high dimensional feature space. Shah et al.

978-1-7281-6916-3/20/$31.00 ©2020 IEEE

[8] extracted the features by using Principal component analysis (PCA), to produce the lower-dimensional feature set from the original text and get rid of irrelevant feature space. Also, they have used information gain and ambiguity measures for feature selection to improvise the model. Dwivedi et al. [9] did a comparative study on a rule-based model (RBM) feature and lexicon features, for the optimization of sentiment classification. Kumar, S et al. [10] worked on the N-gram model sequences used for feature extraction includes unigram, bigram and trigram based on the importance of context, for varying analysis of co-occurring word within a given window. The methodology demonstrates to understand the sequences of words from a given sample text. In [11] author proposed a Term Frequency-Inverse Document Frequency (TF-IDF), for feature extraction trained by Support vector machine (SVM) algorithm that produced a satisfactory classification. Manchanda et al. [12] compute a dynamic algorithm programming, for segmenting a text from the labelled document with the help of distant multi-class supervised approach. Jo et al. [13] proposed a text segmentation with K Nearest Neighbors, by classifying that the sentence or paragraph belongs to a specific group, which considers the similarity between attributes by calculating similarity among feature vectors. This analysis found that similarity features represent an important role in segmenting various information from the document. Hakak et al. [14] represent the survey of single-pattern exact string matching algorithms and the working process of different string matching algorithms, to identify proper string matching algorithms depending on their application and complexity of the problems. Ensemble learning combines more than one machine learners into one ensemble learner according to ensemble policy. Ensemble learning algorithms try to produce a set of learners from training data set and integrate these learners to deal with the same task. Zhang et al. [15] uses ensembling techniques like Random forest (RF), Bagging and Ada Boost classifiers for imbalanced text classification and also proposed sampling techniques like under-sampling, bootstrap re-sampling used to improve the classification performance. Goudjil et al. [16] approached the active learning multi-class support vector machine (SVM), which requires minimal labelled data gives the better model, where the algorithm chooses the data to learn by selecting the samples considered to be the most informative. Xu et al. [17] proposed a complex model using Extreme gradient boosting (XGB), for the recommendation engine to predict the probability of every sample and taken the appropriate threshold to decide the label. So, they converted the recommendation problem into a binary classification problem and extracted the features from consumers behaviour history to predict the target user. Their research shows that it can handle enormous records in a short time and perform well.

In our research study, we have proposed two levels of classification techniques to extract segment information from the resume. First, a classification model has been built to predict a text line in the resume is a heading or not. The heading is then classified as different heading categories. Finally implemented a method that extracts the segment under each heading. The paper is set out as follows: - The processing of data is dealt with in Section II. Section III explains the approach proposed to resume segmentation. The next segment focuses on performance and technique evaluation. Finally, we conclude the paper.

II. Data Preparation

A. Data Collection

In our recruitment system, all the resumes gathered from different sources for various requirements are stored in a database. Resumes were stored in various file formats such as DOC, DOCX, PDF, TXT, HTML, and so on. We have considered MS Word and PDF files for this experiment. 400 resumes were randomly picked from the recruitment database and considered this as a base dataset for our experiments. Extracting text from resume is a challenging task. We've explored python packages such as pypdf2, docxt2txt, OCR to extract the text from a resume, but it leads to some issues including a line breaker when beginning of bold word starts, incorrect text and table alignment. All the files have been processed first to extract text and tables from the resumes. Pydocx, an open-source library has been used to parse docx files and Pdfminer, a python based open-source library used to extract text and tables from pdf files. The extracted text further used to train classifiers for heading prediction.

B. Data Labelling

Data labelling refers to the method of assigning data point labels, to make the information suitable, for the training of supervised machine learning models. Each row of text extracted from resumes is labelled manually by experts. Label 1 has been assigned to all the text, which is a heading and 0 for not-a-heading. 26,092 line of text from resumes have been labelled. Table I shows the example of labelled data points for a resume.

TABLE I. Sample Labelled Data

Text	Target class
SUMMARY OF EXPERIENCE	1
Highly skilled at building teams, maintaining a healthy and conducive environment for productivity and Team work.	0
SKILLS PROFILE	1
Platform/Technologies: SAS, SQL, PL/SQL Basics, Unix Commands and Basic level Shell Scripting	0
Learning Technologies: R, Tableau and Python, Data Science Concepts	0
PROJECTS SUMMARY	1
Project Names: AG Insurance	0
Description: The customer is an insurance-banking firm involved in developing marketing insurance and services for self-employed persons, private customers and small businesses based on specifications by the business analysts as well as programming using SAS Base.	0

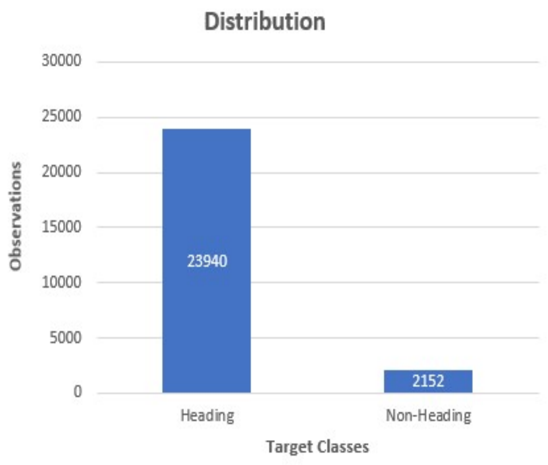

Fig. 1. Distribution of target class

Distribution of class ratio shown in Fig. 1. If the text within the data was a heading, the label was set to 1, otherwise 0. Labelling text is one of the most critical steps because the model's performance depends on the quality of the data.

C. Data Pre-processing

The process of transforming raw data into usable training data is referred to as data pre-processing. Inadequate resumes are dropped from the sample, which could not provide the proper alignment. It means usually resume contains text in different format like list, table and combination of both. We have excluded the table format resumes for classifier model which didn't give a proper alignment. Raw data is always unacceptable in the real term, and data cannot be sent through a model because it causes certain errors. So, it will be easy to pre-process the data before fitting through a model. Then, 333 unique resumes have been finalized from the selected sample, which is processed further to extract text lines. Next, we dropped missing values and the text lines which contain only special characters, to reduce model complexity.

D. Feature Extraction

Different types of features are extracted and analyzed to collect meaningful insights from the text, to differentiate the heading and not-a-heading. Some features are described in the following sub-sections.

1) *Word Count:* A total number of words in the text. The basic intuition behind this feature generally, heading contains a lesser amount of words than the not-a-heading.

2) *Length of text:* Number of characters in the text.

3) *Text ends with symbol:* Text line end with any special pattern or symbol. Usually for headings word will end with some patterns like a colon (:), hyphen (-) or combination of both (:-) and not-a-heading or contents will end with a dot (.) to differentiate it.

4) *Average word length:* Average word length in each text, used to calculate the average word length of each block. This can also potentially help us in improving the model.

5) *Stop Words:* Number of stop words.

6) *Numeric:* Number of numeric values

7) *Special Characters:* Number of Special Characters

8) *Text Case Feature:* This feature describes whether all the words in the text is in upper case, lower case and camel case (first letter of all words in uppercase) or not. Since people use camel case and upper case for headings, mainly to make a difference from heading or not-a-heading.

9) *Part of Speech Feature (POS):* The recurrence for every POS label is determined and used to compute the frequency of every POS tag in the content for every datum point. These frequencies lead to potential highlights for the model.

10) *Similarity Features:* We considered the most repeated words present in the combination of heading text and created a feature for each word taken. Then, we calculated the cosine similarity for each word feature with the text present in each block of information.

E. Feature Analysis

All the features were analysed to observe the impact of features on the target class. For example, average word count is more in class 0 compared to class 1 shown in Fig 2 because words in not-a-heading will be more than heading. Then the probability of text ends feature is more in class 1 than class 0 shown in Fig 3 because unlike not-a-heading words, heading mostly represents the symbols like colon, hyphen and combination of both at the end.

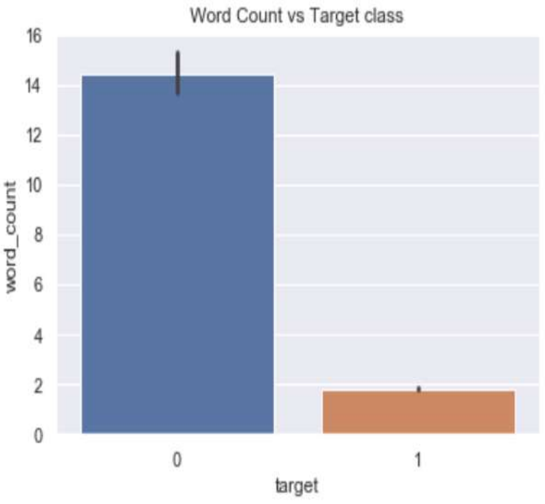

Fig. 2. Word count impact on target class.

F. Feature Transformation

Standardization is an important technique mostly preferred as a pre-processing step before getting into Machine Learning models. Most of the algorithms perform well when variables are relatively on a similar scale or close to normal distribution. Some of the extracted features are highly skewed, In order to, normalize the skewed feature, we have used Robust Scaler technique. It transforms the variable vector by subtracting the median and then dividing by the interquartile range (75% value – 25%). It also handles outliers which require scaling the data and achieving normal distribution. Then, the categorical features were transformed by applying one hot encoding technique, including bold and text ending with the special pattern. It would result in a dummy variable trap, as other variables can easily predict the outcome of one variable. But the dummy variable trap leads to multicollinearity problem because of dependency between the independent variables. So, we dropped one of the dummy variables from each categorical to get rid of this problem.

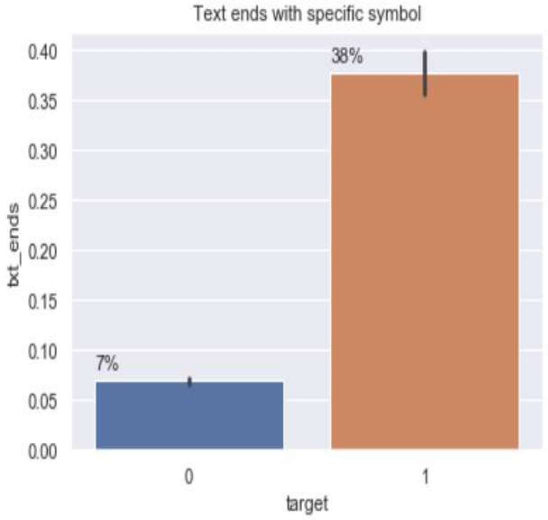

Fig. 3. Impact of text end with special symbols on target class.

G. *Creation of Training and Validation data*

In general, the data is divided into two parts; train and test using the stratified sampling technique, with the same ratio of two labels distributed in both the sets. So, the model can be generalized well on unseen data. In other terms, it can predict better accurate results on its adjustment of internal parameters when it was trained and validated. The total number of observations taken in train and test shown in Fig 4.

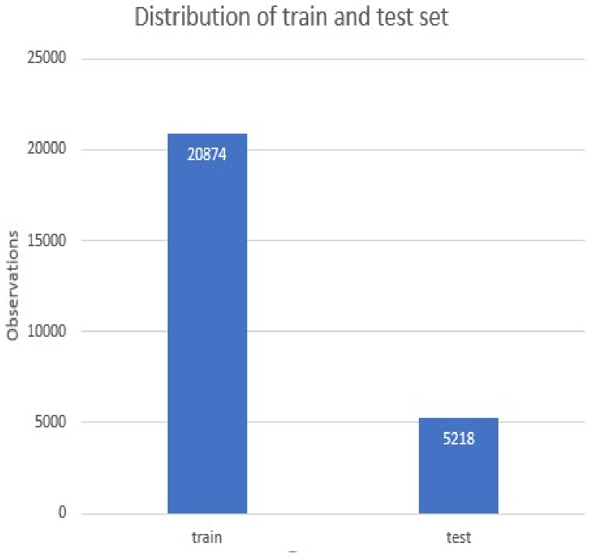

Fig. 4. Number of observations for train and test data.

III. Methods

In general, resumes are classified into multiple sections like objective, summary, skillset, education, experience, project and personal details. Under each heading, the content will be associated with a detailed description. All the sections are separated based on title heading. We considered each segment as a section of a resume. To extract such segment's details, we have used multi-level classification and retrieve information under that segment. The whole process has three main steps. a) heading prediction – that predicts a line is heading or not, b) segment classification – that maps a heading to a generalized segment label, and c) segment extraction -that extracts segment details.

1) Heading Prediction: Classifier model is fitted on labelled data to allow text from the resume and differentiated the heading and not-a-heading based on training data. For model training, we have extracted 61 features to analyze the relationship by learning the function that maps the input(variables) to the output(target). We've also split the dataset into train and testing for model building. Then, we used a scikit-learn module to continue with the classifier algorithm. Multiple classification models have been built using various techniques, such as K-Nearest Neighbors (KNN), Support Vector Machine(SVM) with RBF kernel and ensembling techniques like Bagging, Random Forest (RB), Gradient Boosting (GB), Extreme Gradient Boosting (XGB), Ada boost and LightGBM. Finally, The best classifier was chosen for the heading prediction by evaluation metrics. But this heading prediction differs when the entire resume represents in table format as mentioned the difficulties in data preprocessing section. So, instead of classifier model similarity features are used to identify the heading based on a threshold value. So, we proposed the rule if the resume in list or combination of list and table format will predict the headings by the classifier model and prediction based on a threshold value using similarity score for the table format.

2) Segment Extraction: After the heading prediction, we segmented the different section of information from resume by rule-based technique. The proposed rule identifies the start of the heading, which was mentioned in a predicted label as (1) and it will extract all the information until the next heading appears. Therefore, all the extracted information is considered as the details of the previous heading. Then, we created a dictionary to store all section information in the key-value pair for every resume. Here key holds the title or heading and its respective content present in the value. This is because key should be unique as well as heading imparts the different information with distinctive title and not-a-heading as a value pair of each key. Table II shows the example of the dictionary contains a map of all respective contents to its unique title from the predicted resume. By this technique, we successfully converted the unstructured information to structured format from resume. This dictionary key will be pass into the next step for the specific segment extraction.

TABLE II. DICTIONARY OF HEADING AND ITS DETAILS

Key	Value
OBJECTIVE:	A post graduate who is seeking a challenging position, with a strong emphasis on Data science Technology, where I can use my abilities and explore the best of skills and acumen to become a valuable asset to the organization.
PROFESIONAL SUMMARY:	Have knowledge about working and architecture of HDFS', Worked and contributed with the team in the project.
TECHNICAL SKILLS:	Programming Languages: JAVA, SQL(BASIC), R, PYTHON, Operating System: Windows, Linux
EXPERIENCE:	Tata Consultancy Services Internship 20 November 2017 - 19 March 2018 Chennai, Siruseri campus, Got initial training in Scala and Kafka.
AWARDS & ACHIEVEMENTS:	Won inter house debate competitions in school. Served as organizer Jet club-2014 in college.

3) Segment Classification: We have categorized 20 classes for all the major headings in resumes. We implemented the rule to extract the specific skill information from the dictionary, based on approximate string matching algorithm fuzzy-matching. Since fuzzy search will look for the outputs, that misspelt or differently punctuated words. Unlike rigid search looks for the exact instances. Each key-value information is passed through the partial string-matching algorithm to match with all possible names of the skill. At that time, the fuzzy match algorithm calculated the partial string match, by using Levenshtein Distance metric with configurable parameters like maximum allowed distance, substitutions, deletions and insertions. Each key was ranked, based on the distance range and matched with a possible number of elements in a skill list. Therefore, the higher rank considered as skill details based on the distance score. Table III shows the output of skillset information. We have explored only the skillset category, although this can apply to other section categories.

TABLE III. EXTRACTED SEGMENT FOR SKILLSET

Segment Category	Output
Skillset	Programming Languages: JAVA, SQL(BASIC), R, PYTHON, Operating System: Windows, linux

IV. RESULTS

The classification models were evaluated using performance metrics like Accuracy, F1 Score, Precision, Recall and Area Under Curve (AUC). These performance metrics can be calculated using true positive (TP) value, false positive (FP) value and false-negative (FN) value. TP is the case when a particular text is correctly identified as a heading. If a text is identified as a heading, but it is a not-a-heading then such case is considered as FP. FN is the case when a heading is detected as not-a-heading. F1 score will be calculated by using the weighted average of precision and recall. Equation (1) and (2) describes the formula for the calculation of precision and recall. F1-score is calculated based on (3). AUC ROC graph will show the overall performance of the classification model at all thresholds. We have chosen the F1 score metric instead of accuracy for the assessment of heading classification, as the dataset is imbalanced.

$$\text{Precision} = TP/(TP+FP) \quad (1)$$

$$\text{Recall} = TP/(TP+FN) \quad (2)$$

$$F1 = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (3)$$

F1 score, precision and recall have been calculated using the test dataset and analyzed the performance of all the classifiers used for model training as defined in Table IV. XGBoost outperforms the other classifiers used. Finally, we considered XGBoost as the classification model for the prediction of a text is heading or not. Moreover, we've separately measured the impact of features for the XGBoost model. Table V shows the confusion matrix to visualize the actual and predicted values from the model. Fig. 5, shows the ROC curve and AUC for the XGBoost. Also, we tried to tune the classification model using the probability threshold value of predicting heading and not-a-heading using XGBoost model. In our experiment, precision reduced when the threshold adjusted below 0.5 and recall decreased above 0.5. So, it results in lower F1 score and needs to focus on both values. Therefore, the default probability threshold value fixed to 0.5 gives the best performance of the model. Fig. 6 describes the important features with predictive power. The number of words present in the text has the highest impact. We have also tested the performance of skillset extraction using test data which has 33 resumes with PDF and MS Word files. We have successfully extracted the skillset from 28 resumes, which gives 85% of accuracy.

TABLE IV. PERFORMANCE OF CLASSIFIERS

Classifiers	Performance Measures in %		
	F1-Score	*Precision*	*Recall*
KNN	0.846	0.872	0.823
BAG	0.875	0.920	0.834
RF	0.891	0.902	0.881
GB	0.893	0.896	0.890
AB	0.876	0.876	0.876
XGB	0.901	0.914	0.898
SVM	0.85	0.899	0.811

TABLE V. Confusion Matrix For XGB

Confusion Matrix	**Actual Vs Predicted**	
	Predicted Not-a-Heading	***Predicted Heading***
Actual Not-a-Heading	4752	36
Actual Heading	48	382

V. Conclusion

We have extracted various features from text and built a classifier to predict whether a text line in a resume is heading or not. Our research shows that XG boost outperforms the other classifiers in predicting headings. After the prediction by using fuzzy search technique based on the edit distance, we successfully extracted the skillset information from both the PDF and Word files. Manually annotated data created a more realistic view of the data generation process. So, it generalized well on unseen resumes. In future, we will enhance our work by including all formats of different files of resume and optimize the performance of the classifier, to improve the accuracy in predicting and retrieving all the 20 segments from the resume. We also want to build a classification model for the prediction of the category of each heading. This extracted skillset, experience and education can be used to recommend resumes for a job requirement.

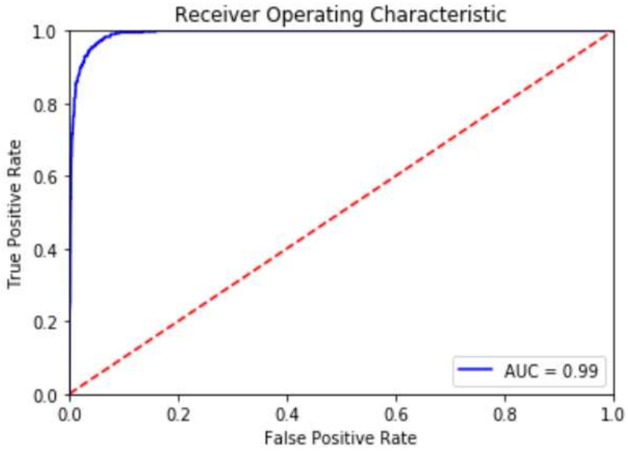

Fig. 5. ROC and AUC Based on XGB Model.

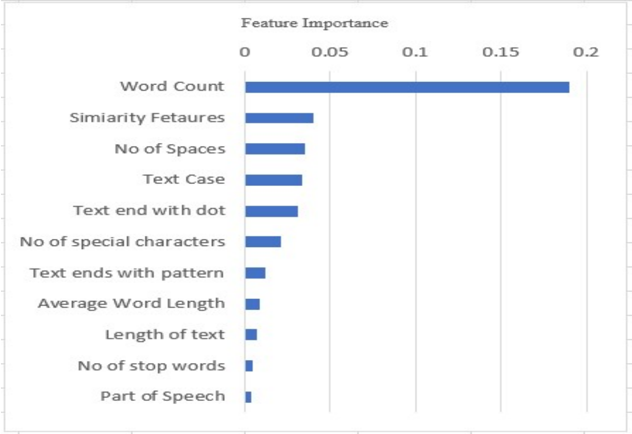

Fig. 6. Feature importance of XGB Model.

References

[1] Chif, E. S., Chifu, V. R., Popa, I., & Salomie, I. (2017, September). A system fordetecting professional skills from resumes written in natural language. In 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 189-196). IEEE.

[2] [2] M. Zhao, F. Javed, F. Jacob, and M. McNair, SKILL: A System for Skill Identification and Normalization, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 4012-4017, 2015.

[3] Chen, J., Zhang, C., & Niu, Z. (2018). A two-step resume information extraction algorithm. Mathematical Problems in Engineering, 2018.

[4] Sanyal, S., Hazra, S., Adhikary, S., & Ghosh, N. (2017). Resume Parser with Natural Language Processing. International Journal of Engineering Science, 4484.

[5] Bernabé-Moreno, J., Tejeda-Lorente, Á., Herce-Zelaya, J., Porcel, C., & Herrera-Viedma, E. (2019). An automatic skills standardization method based on subject expert knowledge extraction and semantic matching. Procedia Computer Science, 162, 857-864.

[6] Ayishathahira, C. H., Sreejith, C., & Raseek, C. (2018, July). Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing. In 2018 International CET Conference on Control, Communication, and Computing (IC4) (pp. 388-393). IEEE.

[7] Kumar, R., Agnihotram, G., Naik, P., & Trivedi, S. (2019). A Supervised Method to Find the Relevance of Extracted Keywords Using Deep Learning Approaches. In Emerging Technologies in Data Mining and Information Security (pp. 837-847). Springer, Singapore.

[8] Shah, F. P., & Patel, V. (2016, March). A review on feature selection and feature extraction for text classification. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 2264-2268). IEEE.

[9] Dwivedi, R. K., Aggarwal, M., Keshari, S. K., & Kumar, A. (2019). Sentiment analysis and feature extraction using rule-based model (RBM). In International Conference on Innovative Computing and Communications (pp. 57-63). Springer, Singapore.

[10] Kumar, S. S., & Rajini, A. (2019, July). Extensive Survey on Feature Extraction and Feature Selection Techniques for Sentiment Classification in Social Media. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[11] Dadgar, S. M. H., Araghi, M. S., & Farahani, M. M. (2016, March). A novel text mining approach based on TF-IDF and Support Vector Machine for news classification. In 2016 IEEE International Conference on Engineering and Technology (ICETECH) (pp. 112-116). IEEE.

[12] Manchanda, S., & Karypis, G. (2018, November). Text segmentation on multilabel documents: A distant-supervised approach. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 1170-1175). IEEE.

[13] Jo, T. (2017, January). Using K Nearest Neighbors for text segmentation with feature similarity. In 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE) (pp. 1-5). IEEE.

[14] Hakak, S. I., Kamsin, A., Shivakumara, P., Gilkar, G. A., Khan, W. Z., & Imran, M. (2019). Exact String Matching Algorithms: Survey, Issues, and Future Research Directions. IEEE Access, 7, 69614-69637.

[15] Zhang, D., Ma, J., Yi, J., Niu, X., & Xu, X. (2015, August). An ensemble method for unbalanced sentiment classification. In 2015 11th International Conference on Natural Computation (ICNC) (pp. 440-445). IEEE.

[16] Goudjil, M., Koudil, M., Bedda, M., & Ghoggali, N. (2018). A novel active learning method using SVM for text classification. International Journal of Automation and Computing, 15(3), 290-298.

[17] Xu, A. L., Liu, B. J., & Gu, C. Y. (2018, July). A recommendation system based on extreme gradient boosting classifier. In 2018 10th International Conference on Modelling, Identification and Control (ICMIC) (pp. 1-5). IEEE.