AUTOMATED TOOL FOR RESUME CLASSIFICATION USING SEMENTIC ANALYSIS

Suhas Tangadle Gopalakrishna[1] and Vijayaraghavan Varadharajan[2]

Infosys Limited, Bengaluru, India

ABSTRACT

Recruitment in the IT sector has been on the rise in recent times. Software companies are on the hunt to recruit raw talent right from the colleges through job fairs. The process of allotment of projects to the new recruits is a manual affair, usually carried out by the Human Resources department of the organization. This process of project allotment to the new recruits is a costly affair for the organization as it relies mostly on human effort. In the recent times, software companies round the world are leveraging the advances in machine learning and Artificial intelligence in general to automate routine tasks in the enterprise to increase the productivity. In the paper, we discuss the design and implementation of a resume classifier application which employs an ensemble learning based voting classifier to classify a profile of a candidate into a suitable domain based on his interest, work-experience and expertise mentioned by the candidate in the profile. The model employs topic modelling techniques to introduce a new domain to the list of domains upon failing to achieve the threshold value of confidence for the classification of the candidate profile. The Stack-Overflow REST APIs are called for the profiles which fail on the confidence threshold test set in the application. The topics returned by the APIs are subjected to topic modelling to obtain a new domain, on which the voting classifier is retrained after a fixed interval to improve the accuracy of the model.Overall, emphasis is laid out on building a dynamic machine learning automation tool which is not solely dependent on the training data in allotment of projects to the new recruits. We extended our previous work withnew learning model that has the ability to classify the resumes with better accuracy and support more new domains.

KEYWORDS

Ensemble learning, Dynamic Classification, Machine Learning, Resume classifier Application, Topic Modelling

1. INTRODUCTION

Recruitment in the Information Technology sector has seen an exponential increase in recent times. Companies round the globe recruit thousands of young talent, right from the college every year through campus fairs. Allotment of the projects to the new recruits is one of the pain points for the organisation. The Human Resources(HR) team is entrusted with the responsibility of allocation of projects to the new recruits, which is a manual affair. Manual allotment of projects to the new recruits by opening and analysing the resumes one by one is a tedious and a redundant process. In this paper, we have designed and developed a mechanism which allots the projects to the new recruits of the organisation by considering the skill sets, interests and work experience mentioned in the resume of the candidates. The world of Artificial Intelligence [AI] and Machine Learning [ML] has grown significantly in recent times. The availability of large amounts of data brought about by advancements in technology which has made the internet cheap and accessible to previously inaccessible regions of the world has contributed to a great increase in the performance of the ML models in recent times. Software companies around the world are

DOI : 10.5121/ijaia.2019.10102

exploiting the advances in ML to drive automation and increase their productivity in areas which relied mostly on manual human labour. Excerpts from a survey by IBM [9] reveal that there is roughly about 2.5 Exabyte which corresponds to about 2.5 billion gigabytes (GB) of data in the world. The advent of big data tools like Hadoop, Spark has enabled the software companies to store and analysesuch great amounts of data and build ML models to drive automation and increase their productivity. The companies recently are on a spree of recruitment in different and new areas. The current MLmodels and tools for resume classification in the market are known for classification of resumes into few popular domains of any particular field. Most of the popular resume classifier applications for computer science classify a resume into 'AI', 'Computer Networks', 'Programming Languages', 'Web development' and a few other popular research domains in Computer Science. Hence, for profiles which do not fall under any of such domains, the classifier arbitrarily classifies the profile into any of the above domains. The objective of our research is to provide a solution for the above problem by building a dynamic machine learning model which does not solely depend on the training data to classify a given resume. This is achieved by building a model which is in a constant learning mode. For those resumes with specializations that has not been included in the training set, the tool extracts the features out of the resume, analyses and passes the knowledge extracted from the profile to a REST API trained on the stack-overflow dump for additional information on the category of domain to which the profile belongs to. The Stack-Overflow APIsare trained on the dump which has been posted and answered by the users in its forum. The identity of the users is kept secret. Association Rule Mining algorithm is done on the data dump in order to obtain the related topics for a given particular topic based on the responses from the users for the question related to the topic. The response from the Stack-Overflow APIs is then subjected to topic modelling to obtain a suitable specialisation name for the resume in consideration. The new specialisation is further added to the list of domains on which the ML model was trained initially. The ML model is then retrained with the additional new domain added to the list to improve the accuracy of the prediction. In this way, we are able to eliminate the dependency of initial training set for accurate prediction of the profile into a suitable domain, a major elimination of the present-day resume classifiers in the market.

2. RELATED WORK

There has been some great work in the field of semantic analysis of text and its categorization in recent times with the recent advances in the field of AI and ML [10]. The authors in [1] analyse and compare the different ML techniques that can be used for the classification and text categorization [5]. But the limitation of the approach is that the list of categories to which the documents have to be classified have to be listed during the training phase. Because of the above limitation, the learning model can only classify the resume in the domains on which it has been trained initially. V Ram and Prasanna have emphasized the use of neural networks in the field of text categorization [2]. They have highlighted that by feeding the neural networks with sufficient data, the model would be successful in predicting the category of the text data with a greater accuracy. Yieng Huang and Jingdeng Chen have argued for the use of Deep neural networks for text classification [3]. Deep neural networks are variants of neural networks which have multiple layers of neurons in the intermediary stage before predicting the classification for the text data. With the availability of huge amounts of data, the accuracy of the deep learning models is greater than the traditional single layer neural networks. But in all the cases, there has been a sole dependency on the training data to predict the classification for a given text. Our approach aims at

the elimination of this dependency on the training set for the effective classification of the resume into a suitable domain, and further allocation of project for the candidate in the particular domain.

3. OUR APPROACH

The resume classifier application consists of two main modules as shown in Figure 1. The modules are: a) Natural Language Processing Pipeline (NLPP) and b) Classification module. The profiles of the new recruits are fed as an input to the NLPP by the HR team of the organisation. The module is responsible for eliminating the unnecessary information from the resume and provide only the required data in the form of tokens which could aid in the process of allotment of projects to the classification module. The classification module analyses the list of tokens in order to classify a resume into suitable domain. The application plots a graph depicting the relevance of the candidate with respect to various domains depending upon the interests, work-experience and expertise mentioned by the candidate in the resume. Since, the relevance of the candidate for different domains is provided to the HR team of the organisation, the candidate can be allotted a project in an alternate domain he is interested in, if the domain he is most suitable does not have any vacant jobs. The application automates the task of project allocation, thereby eliminating the tedious and redundant affair of opening and analysing the resumes manually by the HR team of the organisation. In this extended version of our paper, we have added a new classifier (K-nearest neighbours) in the list of individual classifiers constituting an ensemble based voting classifier, thus improving the efficiency of the application. The updated learning model has the ability to classify the resumes with better accuracy and support more new domains apart from that listed in the previous work [16].

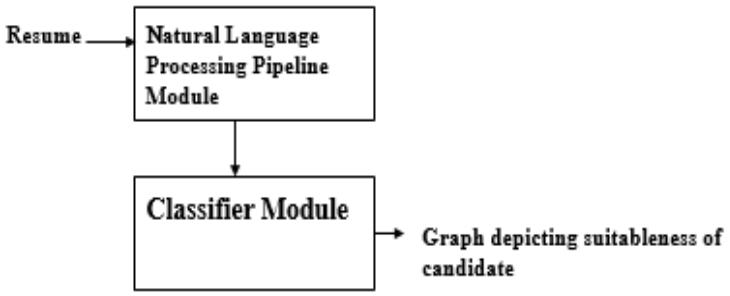

Figure 1. Block Diagram

4. NATURAL LANGUAGE PROCESSING PIPELINE

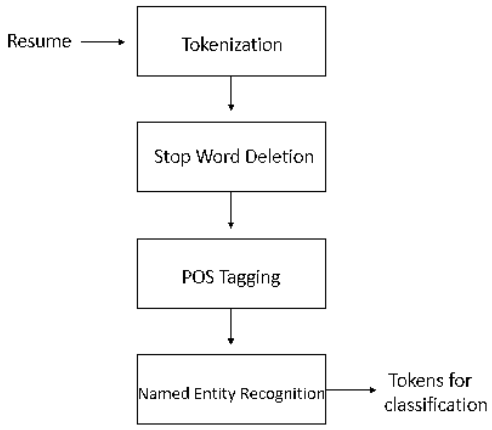

Figure 2: Flow chart of NLPP module

Figure 2 gives a detailed overview of the entire process of NLPP. The new recruits are allotted projects based on the content present in their respective resumes. The data in the resumes of the candidates are subjected to a NLPP in order to obtain only necessary and relevant details. Figure 3 depicts a portion of sample machine learning profile.

- Sentiment analysis and classification using tensorflow in MIT
- Fraud detection using neural networks
- Twitter sentiment analysis
- Spam filter using machine learning algorithms
- Proficient in deep learning tools like keras, theano

Figure 3: Portion of a sample machine learning profile

4.1. Tokenization

The text content in the profile are segmented to obtain tokens. The delimiter for tokenization is the space character. Tokenization involves breaking up of the portion of the ML resume into tokens as depicted in Figure 4. The sentence in the resume, "sentiment analysis and classification using tensorflow in MIT", is converted into a list of tokens comprising of individual words "sentiment", "analysis", "and", "classification", "using", "tensorflow", "MIT".

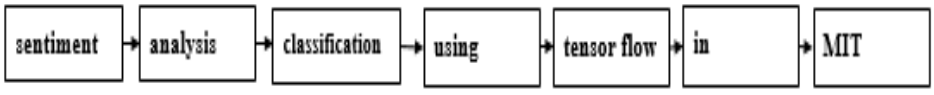

Figure 4: Tokens

4.2. Stop Word Deletion

Usually the resumes of the candidates are filled with redundant words such as 'is, and' etc. Such terms are called stop words. The removal of such simple stop words from the tokens obtained in the previous step resulted in a 28% rise in the efficiency of the classification model. The elimination is important because inclusion of stop words in the training set would result in false learning by the classifier, which would limit the efficiency of classification. The tokens, generated in the last step involve a stop word "and". The word does not hold significance in the classification of the resume. Hence, the word is deleted from the tokens as shown in Figure 5.

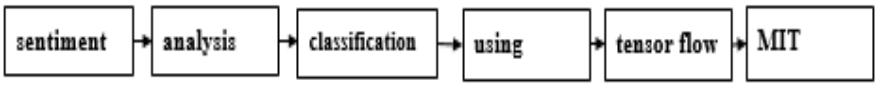

Figure 5: Tokens after stop word deletion

4.3. Parts of Speech Tagging

POS tagging is the next step followed in the NLPP. The step involves tagging the Part of Speeches to each of the tokens obtained after eliminating the stop words. English language has 8 different part of speeches: Verb, Noun, Adjective, Adverb, Pronoun, Preposition, Conjunction, Interjection. The tools and technologies used, Projects undertaken by the candidate is a noun or a pronoun. Since, the allotment of domain to the resume is dependent mainly on these features, only the tokens labelled as nouns and pronouns (NNP) are considered for the next step of NLPP. This step reduces the computation as the number of words considered for classification is reduced greatly. The POS tagging [7] of the tokens is shown in Figure 6. Only the tokens labelled as nouns move to the next stage in the pipeline.

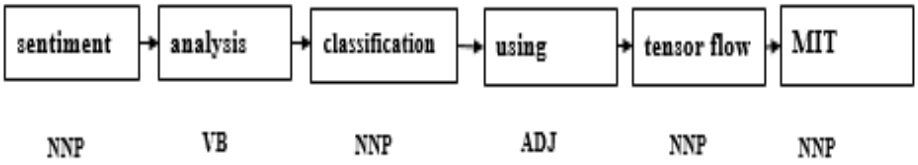

Figure 6: Tokens with POS tagging
NNP: Noun or a pronoun
ADJ: Adjective
VB: Verb

4.4. Named Entity Recognition

All the tokens labelled nouns and pronouns are subjected to Named Entity Recognition [8]. The tokens include the names of the candidate, educational institutions and place names. These tokens are eliminated in the present stage of the pipeline by identifying the candidate names, organizations and place name tokens, which are of trivial importance for allotment of a project to the candidate. This is the final step of the pre-processing pipeline. The tokens emerging out of this step are considered by the classifier for classification. The tokens which are recognized as name, place or organization in this step are eliminated as shown in figure 7. Only the contents shown in Figure 8 move to the classification module.

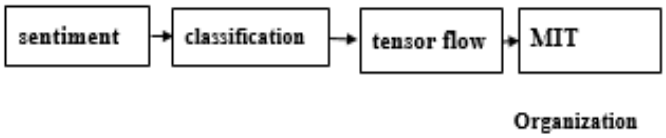

Figure 7: Tokens with Named Entity recognition

The process is repeated for every sentence in the resume portion shown in Figure 3. The final set of tokens are then forwarded to the classifier module as shown in Figure 8**.**

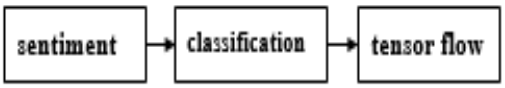

Figure 8: Output from NLPP

5. CLASSIFICATION MODULE

Once the NLPP module generates the set of tokens from the text content in the resume, the set of tokens are then passed to the next stage of the application: Classification module. The classification module is responsible for the classification of the tokens into suitable domain, so that a suitable project can be allotted to the candidate in the domain. The flow chart of the classification module is illustrated in Figure 9. The classification module comprises of an ensemble learning based voting classifier consisting of 5 individual classifiers i) Naïve Bayes ii) Multinomial Naïve Bayes, iii) Linear SVC, iv) Bernoulli Naïve Bayes v) Logistic Regression [15]

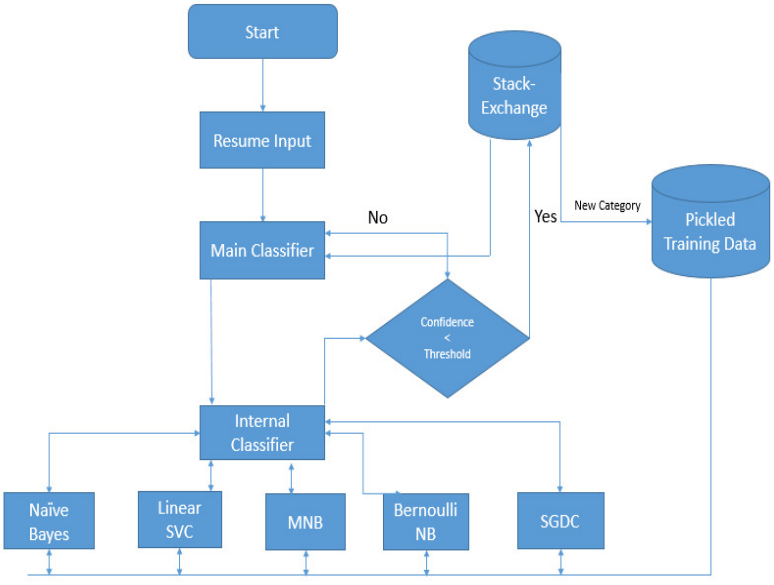

Figure 9: Flow chart of classification module

The individual classifiers which constitutes the ensemble learning based voting classifier are the commonly used learning models in the field of machine learning. All the individual classifiers were trained on 30,000 employee profiles who were previously allocated projects based on their interests, work-experience and expertise mentioned in the resume. The list of domains in which the employees were previously allocated projects are listed in Table 1. The dataset of 30,000 employee profiles was split in a 9:1 ratio wherein 27,0000 profiles formed the basis of training set while 3,000 profiles were used to test the accuracy of the classifiers. The performance of each of the individual classifier was recorded as seen in Table 2.

Table 1: Initial Domains to be mapped with the profile

Domains
Artificial Intelligence
Computer Architecture
Computer Graphics
Databases
Distributed Computing
Computer Networks
Web Technologies
Cloud Computing

Each classifier after training predicts the test data based on the learning from the training set. Efficiency of the classifier is the percentage of number of right predictions by the classifier the test data. The efficiency of each individual classifier is shown in the Table 2.

Table 2: Efficiency of individual classifiers

Name of the Classifier	**Efficiency of prediction in %**
Naïve Bayes	79
Linear SVC	83
MNB	91
Bernoulli NB	89
Logistic Regression	81
K- nearest neighbours	80

K-nearest neighbours and Naïve Bayes are the classifiers with the least efficiency on the training set. The K-nearest neighbours algorithm classifies the data from the test set into a domain based on the distance of the data point from the centroids of the different clusters. Since K-nearest neighbours and Naïve Bayes algorithms are the least efficient classifiers on the training set, they are given the least votes to classify the profile to a domain. The classifiers are allotted votes based on their respective efficiencies as shown in Table 2. This ensures higher influence of the classifier having greater efficiency in categorizing the tokens to a domain than the classifier having less efficiency, while mapping the profile to a domain. The number of votes given to each classifier is governed by equation 1.

$$\text{Number of votes to a classifier X} = \frac{Efficiency\ of\ classifier\ X}{Efficiency\ of\ classifier\ with\ least\ efficiency\ in\ training\ data} \quad (1)$$

From Table 2, the least efficient classifier is Naïve Bayes (79% efficient), hence the denominator for the above equation 1, is 79%. Hence, for Linear SVC (83% efficient), the number of votes allotted is 83%/79% = 1.05 votes. The votes allotted to individual classifiers is given in Table 3.

Table 3: Distribution of votes between different classifiers

Name of the Classifier	Number of Votes
Naïve Bayes	1
Linear SVC	1.05
MNB	1.17
Bernoulli NB	1.12
Logistic Regression	1.025
K-nearest neighbours	1.01

Confidence of the main classifier for the selection of domain for the set of tokens is the ratio of the number of votes casted in favour of the majority class to the total number of votes available with the individual classifiers.

$$\text{Confidence} = \frac{\text{Number of votes cast in favor of majority domain}}{Total\ number\ of\ votes} \quad (2)$$

Total number of votes cast by the classifiers are the sum of the votes allotted to the individual classifiers. From Table 3, the total votes allotted = 1 (Naïve Bayes) + 1.05 (Linear SVC) + 1.17(MNB) + 1.12(Bernoulli NB) + 1.025(Logistic Regression) + 1.01 (K-nearest neighbours) = 6.375. If Naïve Bayes, Linear SVC, Bernoulli NB and Logistic Regression choose machine learning domain for the set of tokens from Figure 8, while the rest of the classifiers chose distributed computing for the same set of tokens. Machine learning domain has a total vote share of 4.195, while distributed computing has a share of 2.18. Hence, the machine learning domain with a majority vote share of 4.195 is the domain selected by the main classifier. The confidence of the resume classifier from the equation 2 is given as 4.195/6.375 = 0.658.

Higher confidence of the model depicts higher consensus of the classifiers in selecting the domain and higher the chances of the classifier being correct in the decision to classify the set of tokens into the right domain. There are some instances, when the votes of the classifiers are cast equally on number of different categories. In this case, to break the deadlock, the domain which has been voted by the most efficient classifier is chosen.

6. DYNAMIC CLASSIFICATION

The classification module classifies a particular resume of the candidate to a suitable domain with a confidence value. A threshold value has been set on the confidence value which will prevent the classification module from classifying any resume tokens to a domain if the confidence for the particular classification is less than the threshold values which is set. The threshold value set is 0.55. Hence, for the classification module to successfully classify a resume into a domain,

number of votes for the domain with the maximum votes should be more than half of the total votes available with the individual classifiers.

- Specialised in Microsoft Azure
- Proficient in Amazon Web Services
- Proficient in deployment of applications in cloud
- Amazon Web Service Certification – 2015
- Proficient in SAAS

Figure 10: Portion of sample cloud computing resume

Figure 10 depicts a portion of sample cloud computing profile. Cloud computing does not come under any domain listed in Table 1. The sentence in the resume, "specialized in Microsoft Azure", forms a set of tokens after passing through the NLPP pipeline as shown in Figure 11.

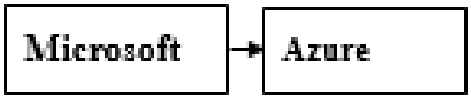

Figure 11.Tokens generated after NLPP

Since the individual classifiers constituting the voting model have no knowledge on these tokens shown in Figure 11, the confidence of the classifier in selecting a domain for the tokens from the NLPP pipeline is less than the threshold value of 0.55. Hence the tokens are passed to the Stack-Overflow REST API. The REST API returns the associated topics for each of the tokens based on the Association Rule Learning [11] of User Question and Answer dump of Stack-Overflow. The topics returned from the REST API is subjected to topic modelling via Latent Dirichlet Allocation (LDA). The topic obtained from LDA is added to the list of domains mentioned in Table 1. The ensemble learning based voting classifier is retrained after the fixed interval including the new domain. The number of votes given to each classifier is modified depending upon the accuracy obtained by the individual classifiers in determining the domain of the resume. In the above example, a new domain of cloud computing is added to the list of domains in Table 1, after performing LDA on the topics returned by the Stack-Overflow API.

7. RESULTS

The result of the resume classifier application is a bar graph which depicts the suitability of the candidate profile for various domains. The x-axis lists all the domains while the y-axis defines the confidence of the model for classifying the candidate profile to a particular domain. Higher value of the y-axis depicts greater suitability of the candidate for the particular domain.

Figure 12 depicts the output from the resume classifier tool for the machine learning profile in figure 3. The classification of the resume into AI domain with a confidence of 0.59 is driven by the recognition of deep learning tools like "tensorflow", "keras", "theano" and machine learning buzzwords like "spam filter", "analysis and classification", "fraud detection" recognised by the resume classifier application in the tool.

The individual classifiers of Naïve Bayes, Linear SVC and Bernoulli NB constituting the ensemble learning voting model classified the resume into AI domain while Multinomial NB and Logistic Regression classified the resume to Distributed computing and Computer Architecture domain respectively as illustrated by the graph output in

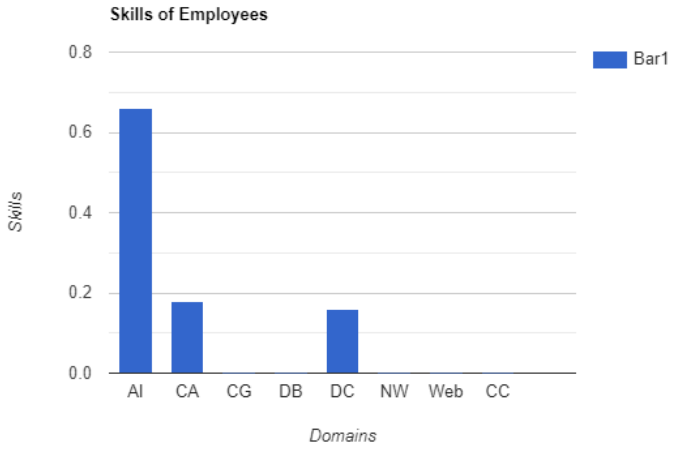

Figure 12: Output graph

AI: Artificial Intelligence

CA: Computer Architecture
CG: Computer Graphics
DB: Databases
DC: Distributed Computing
NW: Networks
Web: Web Technologies

.

The X axis forms the domains in the initial training set, while the Y axis depicts the confidence of the main classifier across various domains. The profile is then allotted a project in the AI domain as classified by the classifier.

$$\text{Confidence for AI} = \frac{\text{Number of votes for AI}}{Total\ number\ of\ votes}$$

$$= \frac{\text{Votes of Naive Bayes+Linear SVC+ BNB+LR}}{Total\ number\ of\ votes}$$

$$= \frac{1+1.05+1.12+1.01}{6.375}$$

$$= 0.66$$

$$\text{Confidence for Distributed Computing} = \frac{\text{Number of votes for Distributed Computing}}{Total\ number\ of\ votes}$$

$$= \frac{\text{Votes of Multinomial NB}}{Total\ number\ of\ votes}$$

$$= \frac{1.17}{6.375}$$

$$= 0.18$$

$$\text{Confidence for Computer Architecture} = \frac{\text{Number of votes for Computer Architecture}}{Total\ number\ of\ votes}$$

$$= \frac{\text{Votes of Logistic Regression}}{Total\ number\ of\ votes}$$

$$= \frac{1.025}{6.375}$$

$$= 0.16$$

The efficiency of the voting based classifier was 91.2%, predicting domains accurately for 2736 out of the total 3000 resumes in the test set, while in 80% of such cases, the confidence of the model was above 0.7. The accuracy without the re-training was 84.2%. Hence, the re-training after topic modelling of the related topics returned from the REST API increased the efficiency of the classifier by 8.3%.

$$\text{Increase in efficiency after retraining} = \frac{efficiency\ after\ retraining - efficiency\ before\ retraining}{efficiency\ before\ retraining} * 100 \quad (3)$$

$$= \frac{91.2-84.2}{84.2} * 100$$

$$= 8.3\%$$

8. CONCLUSION AND FUTURE WORK

The results from the model are encouraging. The resume classifier application is successful in automating the manual task of project allocation to the new recruits of the organisation based on the interests, work-experience and the expertise mentioned by the candidate in the profile. The ensemble learning based voting classifier performs extremely well compared to the individual classifiers while predicting most of the instances of the test data. This is because of the fact that the confidence of the model while categorizing resumes is influenced by the majority of the votes cast by the individual classifiers rather than a single classifier. The retraining of the individual classifiers after a fixed interval to incorporate the classification of new domains returned by the Stack-Overflow API trained on Association Rule Mining of user question and answer dump has shown an increase in the accuracy of the classification of resumes into a suitable domain. In the future, we intent to build an ensemble deep learning model [13] for categorization of the given text content into a suitable domain. We believe that data hungry deep learning models especially Generative Adversarial Network [15] will enjoy greater success in resume classification as harnessing the huge amounts of unstructured and structured data available in the organisation.

9. ACKNOWLEDGEMENTS

We would like to record our appreciation to all the people involved in the research and development of the resume classifier application. We express our gratitude to the InStep team at Infosys for providing us with the requisite dataset involving the resumes of the employees which formed the training set of the classifier.

REFERENCES

[1] A Comparative Study on Different Types of Approaches to Text Categorization, Pratiksha Y. Pawar and S. H. Gawande International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012

[2] Web Document Classification Based on Fuzzy k-NN Algorithm Juan Zhang Yi NiuHuabeiNie Computer and Information Computer and Information Computer and information China.

[3] V Ram, Prasanna, "A unique way of measuring the similarity of the documents using neural networks ,International Journal of EngineeringResearch and Development, Vol.2, no.6, July 2013, pp 397-401.

[4] Yiteng Huang, Jingdong Chen Blind, "Classifying the text using the power of deep learning", International Journal of Engineering and Technology ,Taipei, Taiwan, 2013, pp 3153-3156.

[5] Daniel R and George V, "A Latent Semantic Analysis method to measure participation quality online forums", 2016 IEEE 16th International Conference on Advanced Learning Technologies, January 2016, pp 108-113.

[6] Jongwoo Kim, Daniel X. Le, and George R. "Naïve Bayes Classifier for Extracting Bibliographic Information from Biomedical Online Articles", national Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20894, USA

[7] Natural Language Query Processing Using Semantic Grammar international Journal Of Computer Science And Engineering Vol II Issue II March 2010 pg no 219-233

[8] Natural Language Query Processing international Journal Of Computer application And Engineering Technology and Science IJ-CA-ETS Oct 2009 pg no. 124-129

[9] https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/

[10] Karin Spark Jones, "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, vol. 28, no. 1, pp. 11-21, 1972.

[11] T. Griffiths, M. Steyvers, "Finding scientific topics", Proceedings of the National Academy of Sciences, vol. 101, pp. 5228-5235, 2004.

[12] D. M. Blei, A. Y. Ng, M. I. Jordan, "Latent Dirichlet allocation", J. Mach. Learn. Res., vol. 3, pp. 993-1022, 2003.

[13] Z.-H. Zhou, J. Wu, W. Tang, "Ensembling neural networks: Many could be better than all", Artificial Intelligence, vol. 137, no. 1-2, pp. 239-263, 2002.

[14] Z. Zhu, Q. Chen, Y. Zhao, "Ensemble dictionary learning for saliency detection Image And Vision Computing", Elsevier, vol. 32, pp. 180-188, 2014.

[15] Lin Zhu, Yushi Chen, PedramGhamisi, JónBenediktsson, "Generative Adversarial Networks for Hyperspectral Image Classification",Geoscience Remote Sensing IEEE Transactions on, pp. 2018.

[16] Suhas Tangadle Gopalakrishna, Dr. Vijayaraghavan Varadharajan , "Ensemble learning based voting classifier for dynamic project allotment and classification", 8th International Conference on Artificial Intelligence, Soft Computing and Applications 2018.

Authors

Suhas works at Infosys Limited as a Specialist Programmer as part of Expert Track. His research interests include natural language processing, computer vision, human-computer interaction. He is active in various programming forums including HackerRank, HackerEarth and StackOverflow

Vijayaraghavan is a Principal Research Scientist at Infosys Limited doing research in the field of data analytics, Searchable encryption, Security assessment, Cloud security, Authentication and Privacy protection. He has over 17+ years of experience in the fields of research, industry and aca demia. Prior to that he served as an Assistant Professor and guided many post graduate professional students. He has many granted US patents in key technology areas, published many research papers in International journals and conferences and also served as a Technical Reviewer, Program Committee member and Chair for many conferences around the globe.