|| Volume 5 || Issue 12 || June 2021 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH & ENGINEERING TRENDS
Multidisciplinary Journal
Double-Blind Peer Reviewed Refereed Open Access International Journal

RESUME CLASSIFIER AND RECOMMENDING SYSTEM USING NLP

Muskaan ,Pallabi Mandal, Mehr Broca,Nishidha Ramesh Lekhade,Professor Archana khandekar

MIT World Peace University, School of Engineering

muskaansingh20111@gmail.com,mandalpallabi1999@gmail.com,brocamehr015@gmail.com,nishidhal27@gmail.com,archana.khandekar@mitwpu.edu.in

---***---

Abstract: In the domain of online job recruitment, accurate job and resume classification is vital for both the seeker and the recruiter. We have built an automatic text arrangement that utilizes various techniques like Term frequency-inverse document frequency with Machine Learning and Convolution Neural network for training the model with texts and classifying them into labels and eventually to match their results. Using resume data of applicants, we've categorized them into different categories. Due to the sensitive nature of resume data, we have used domain adaptation. A classifier is trained on a large dataset of job description snippet, which is then used to classify resume data. Despite having a small dataset, consistent classification performance is seen. The primary filter for this type of work is the efficiency the system can provide. We aim to compare the results obtained by various algorithms that are generated using the same data so that the efficiency of each algorithm can be evaluated. From the result, it's evident that character-level CNN gives a far better F1 score compared to other models.

Keywords—*Domain Adaptation, CNN, SVM*

---***---

I INTRODUCTION

In today's fast-moving corporate industry, recruiters often got to undergo vast amounts of resume to research the applicants reliably to make a decision upon the deserving candidates. But it's impossible to stay up with the pace today. So, automated classification of resumes is required to ease out the method. To try to to an equivalent, a bulk of labeled resume data is required, and job openings are divided into a particular number of predefined job categories.

The labeled datasets have job titles and job descriptions, alongside several other parameters. Resumes of an equivalent job category are semantically similar. This similarity is searched for to match the applicants to their respective job categories. The info set springs by extracting real-time resume data of applicants from LinkedIn employing a web scrapping method. Individual LinkedIn URLs were extracted employing a URL extracting platform. The URLs were then processed upon to finally scrape off the resume data within the sort of a CSV file with all the features well distinguished. Convolution Neural network (CNN) has been utilized in several image and text classification applications over the years. We propose a model where a CNN based algorithm is employed to coach a classifier with a dataset of over 1000 applicants. After the training process, it is applied to classify unlabelled resumes. Because it may be a comparison study, various algorithms are used with multiple combinations of features on an equivalent data set, and therefore the results were noted. Algorithms like Support Vector Machine (SVM), Naïve Bayes (NB), Term frequency-inverse document frequency (TFIDF) were initially applied individually then later in an ensemble with another algorithm. Finally, all the various results are compared to seek out the foremost effective method of all of them.

II LITERATURE SURVEY

In [1], Ohma B. Hashemi, Amir Asiae, and Reiner Kraft enhanced the search results by understanding the user query's intent. CNN methodology used for feature extraction. Word2vec used that is pre-trained with filters and feature maps. Over 10000 queries were derived from the logs of a search engine. The rule-based model was taken for comparison. For 14 high-level intent, CNN gives 0.47 as an average F1 score. For 125 class of low intent, CNN gave 0.50 as an F1 score.

In [2], Kevin and Neha classified comments into toxic and nontoxic as well as specify the type of toxicity. Representation of word and character are fed to CNN. Sparse category cross-entropy was used with SoftMax function, whereas for multi-label, a sigmoid function was used with binary cross-entropy. Dataset was gathered from Kaggle of over 159571 labeled samples. Subsampling technique used for binary classification and multi labeled, the type of toxicity was included. A multi-layer perceptron was taken for comparison with LSTM and CNN. LSTM gives better results in word-level analysis with 0.886 F1 scores, and at a character level, CNN proved to be better with 0.8 F1 score.

As demonstrated in [3], Multi-label classification of text documents is used in Czech. The input vector has 1s and 0s. A bag of words is used. Dictionary is built using 20000 words. Forty different kernels used for different sizes. Data was taken from a corpus of 2974040 words from 11955 documents. They concluded that the Proposed topology with output thresholding is efficient in multi-label classification. SoftMax works better.

[4] automation tool used in feature extraction instead of statistical or context properties for the detection of DGA domains. The dataset contains Top 1 million domains, Alexa - NOT DGA, OSINT DGA (about 30 classes) with 750,000 examples of DGA. They achieved a detection rate of 90% with a 1:10000 false-positive (FP) rate—an improvement of twenty times FP improvement over the next best method. The F1 score of 0.9906 was micro-averaged.

In [5], Andrej Karpathy et al. analyzed the semantic content of various applications like search and the summarization of video classification. CNN model was used to get results of image recognition, segmentation as well as detection and retrieval. The model trained by processing approximately five clips a second in case of full-frame networks and 20 clips a second for the multiresolution system. The performance was improved from 43.9% to 63.3%. They concluded that CNN architecture is capable of learning features. Low-resolution architecture speeds up CNN without letting go of the accuracy.

Alex Krizhevsky et al. [6] classified 1.2 million images by training a deep CNN. GPU implementation used to avoid overfitting. The regularization method is known as dropout used. The dataset contains 15 million images from 22000 categories collected from the web and trained the network on raw values of RGB. THE seven CNN model gave good results. The top one is 36.7%, and for the top five, 15.4%.

Using data from HiQ Lab, Eric Boucher, and Clement Renault [7] classified job titles based on Linked Summaries. They grouped the job title in small chunks of categories. Recurrent Neural Network was used and achieved an accuracy of 31.7 percent on 133 classes with a baseline at 16 percent.

In [8], Richard Girshick et al. presented a simple solution for object detection. They achieved a 30% improvement using the Convolution Neural Network algorithm and PASCAL VOC dataset. Ilya Sutskever [9] elaborately describes the nature of the RNN learning problem and also addresses the difficulty of the RNN Learning problem using second-order optimization.

According to [10]when the neural network is trained on the small training dataset, it usually performs poorly. This can be rectified by reducing half of the features on the training set. Random “dropout” gives significant improvements in many benchmark tasks and sets new records for speech and object recognition.

In [11] validated the performance, efficiency of the XGBoost 4j package in a distributed programming environment called Apache Spark for predicting the speed of the wind. XGBoost is used in the training dataset to predict a target variable. In [12], a new method was proposed by combining CNN with Naive Bayes for classifying the generated domains of DGA. Dataset provided by DMD 2018 Shared Task was used for this classification problem

In [13] Sivakumar and D.Rekha used the evolutionary Genetic Algorithm (GA) to maximize the usage of the underwater acoustic bandwidth. The results showed that by reducing the time slots and maximizing the throughput using parallel transmission, UWASN could transmit in average minimal turnaround time. A deep learning method [14] called CNN is used to classify the health-relevant web pages because of its learning power. Also, Character level embedding was used to extract appropriate features using CNN.

III. CONTRIBUTIONS

The main contributions of this work includes the following:

1. The dataset is generated by scraping quite URLs with attributes like name,job description,skill set etc.

2. Convolutional neural network is applied for the primary time to find out the character level features from the given attributes of every LinkedIn profiles.

IV.METHODOLOGY

A. System Architecture

The given system architecture depicts the essential flow of operations within the system of resume classification. the primary is of an applicant uploading his/her resume. This document is raw and unstructured that must be worked upon. Various modules are depicted as a knowledge module, model training, and eventually, testing. Within the data module, the method involved is to access LinkedIn profile URLs from internet and process them with Selenium which will locate the required fields and write the info. In model training, Feature extraction is performed by passing data to embedded layers and convolutions.

B. Data Source

Since the pre-existing dataset wasn't consistent with the standards which were required, we scrapped the info on our own from LinkedIn. It contained multiple fields like job title, description, skills of an individual, location, past experiences, images, and lots of more fields. We used a web API to scrap all the info from LinkedIn using skill-specific keywords to achieve a high level of accuracy within the result section.

Further, we attempt to be more accurate with our result, so we'll propose a primary in school subcategory result generation through our system since there has been extensive development in every category, which is complicated on their own. For this out of those 27 categories, we've confined our look for one category IT (Java developer, Cloud computing, Web developer, Testing field, Python developer, etc.). We've further weakened this field into five more subcategories to get a way of preciseness. During this way, a more efficient result are going to be generated, which might allow the recruiters to get precise data.

LinkedIn doesn't allow scraping through code without permission, but there are specific tools through which it's possible to scrap real-time user data but during a limited way. We used phantom buster API to get data from scratch. There's a restriction of 100 entries per user per day, so we used multiple accounts and combined the info generated. The info generated was stored within the sort of an excel sheet, whichallows various cleaning operations on the data generated. This helped us to refine the info and make it something on which we will work on later.

V.EXPERIMENT

We trained our model using 95% of our resume data and used the remaining 5% to check the trained model. On testing with different algorithms and combinations, results were obtained and were compared with each other. On feature extraction initially, two features, namely, job title and description, were used, and later, it had been done only using the work description to avoid the hints that are by default conceded by the title.

The novelty of our research is that we've used an equivalent data for various algorithms as against various subsets of the dataset. This enables us to realize better accuracy with the algorithms, which way, it's easier to review the behavior of the various algorithms under a particular scenario. The various methods used were CNN and TF-IDF utilized in combination with other algorithms like TF-IDF+NB(Naïve Bayes), TF-IDF+SVM(Support Vector Machine) and TF-IDF+XGB. The results entail precision, recall, F1 score values. Each row within the result table represents each job category (e.g., Python).

In Fig. 1, the Feature Extraction module of the CNN algorithm is shown. This module extracts all the essential features using the CNN algorithm. This module is then fed to the training module, where the model trains for the given data and eventually sent for the prediction to the testing module. In Fig. 2, the model accuracy is small over 60% when Resume Description and Job Title variables are taken under consideration , whereas in Fig. 3, the model accuracy is about 50% when only the Resume Description variable is taken into account . This tells us that the accuracy shown in Fig. 3 is best due to the inclusion of the work Title variable with Resume Description.

CONCLUSION AND FUTURE WORK

We have compared the various state of the art algorithms using similar self-generated data which allowed us to compare them in an unbiased way. The data generation was the most challenging task as the data we generated is very sensitive data and is not accessible for everyone easily. Further, the data was raw and it had more deformities, so we had to clean it up various times. After all the efforts, the results generated were promising. We attained an F1 score of 0.57 for naïve Bayes algorithm, 0.59 for extreme gradient boost (XGB), 0.61 for support vector machine (SVM), and 0.65 in case of convolution neural network. As a result, CNN acquires the maximum result when compared with similar state of the art algorithm; we can very proudly conclude that CNN can also be used to classify text rather than just images and show exceptional results when compared to other text classification algorithms. In the future, Generative Pre-Trained Transformer (GPT -2) algorithm can be algorithms like LSTM (Long Short-Term Memory) can be used to increase the performance on this data.

REFERENCES

[1]Homa B. Hashemi, Amir Asiaee, Reiner Kraft, Query Intent Deduction using CNN, ACM, 2016

[2]Kevin, Neha, Detecting and classifying toxic comments, Stanford University, 2016

[3]Ladislav Lenc, Pavel Kr, Deep Neural Networks for Czech Multi-label Document Classification, CoRR, 2017

[4]Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant, Predicting Domain Generation Algorithms with Long Short-Term Memory Networks, CoRR, 2016

[5]Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, Large-scale Video Classification using CNN, Stanford University, 2014

[6]Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, Large-scale Video Classification using CNN, Stanford University,2014

[7]Eric Boucher, Clement Renault, Job Classification Based on LinkedIn Summaries, Stanford University, 2015

[8]R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation,CVPR, 2014

[9]Ilya Sutskever, Training Recurrent Neural Networks, University of Toronto, 2013

[10]G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing, University of Toronto, 2012

[11]Ramraj, S; Saini, Aeshita; Kaur, Gurleen, Comparative Study of XGBoost4j and Gradient Boosting for Linear Regression, International Journal of Control Theory and Applications

[12]Rajalakshmi, R; Ramraj, S; Kannan, R Ramesh, Transfer learning approach for identification of malicious domain names, International Symposium on Security in Computing and Communication

[13]V. Sivakumar, D. Rekha, Node scheduling problem in underwater acoustic sensor network using genetic algorithm, Personal and Ubiquitous Computing, 2018

[14]R. Rajalakshmi and S. Ramraj, A Deep Learning Approach for URL based Health Information Search, International Journal of Innovative Technology and Exploring Engineering, 2019