National Conference on Technological Innovations in Management, Engineering and Science 2019 April 26-27, 2019

Most Common methods for Job Matching Systems

Nishant Sethi
Dept. of Computer Science Engineering
SRM University, Delhi-NCR, Sonipat, Haryana
Delhi, India
sethi.nishant43@gmail.com

Vasu Gupta
Dept. of Computer Science Engineering
SRM University, Delhi-NCR, Sonipat, Haryana
Delhi, India
vasuonemail@gmail.com

Abstract— All the resumes lying on the wrong table poses a latency in the process of reviewing and selecting of the right candidate and which is not an efficient way at all to be able to bag a job. Now this paper entices the most popular methods and techniques known till now which can be implemented in making a job matching system so smart that the efficiency of matching a right candidate's profile/resume with the actual requirement of the employer hiring for the right designation in the right domain can be increased. This is beneficiary in terms of the job seeker and as well as for the job givers too. A hedge of productivity and potentiality for both the sides can be built. The discussed system is schemed in a manner of approaching the application of text/Information Extraction (IE) from the resumes of the candidates and modelling an entity-based database from them which interprets their profiling in accordance with the job type and it's eligibility by the employer and hence matching the potential ones for recommendation of those candidates' profile to the right desk.

Keywords— Job Matching, Skill Matching, Content-Based Filtering, NER

I. Introduction

E-commerce companies use recommendation systems to provide the best products suited for an individual based on their previous purchases. In the recruitment process, finding the best applicant for a job or getting the best job for any applicant is very tedious. The amount of manual search required to find the very best is not feasible. To excel in this task, job recommendation systems can be built which can help the employer as well as the applicant to find the right choice. This paper discusses about the most common modern methods used for Job matching. These include the process of extracting data from the Resume, classifying the data, identifying the data as well as matching it with a job description.

II. Architecture

This section explains the architecture of the Job Recommendation system. When trying to get the best job for a candidate, The only requirement the system posses is the resume of the applicant in a .pdf format. This ensures that the system is able to read all the text available in the resume. After the text from the resume will be extracted and classification of text will be done next. There are several techniques for this and the most common and accurate ones will be discussed in this

paper. Once the resume is classified, Similar text classification will be done on the job description. Then the most important keywords will be selected and will be intersected in both the documents. The most number of keywords matching will be the most ideal match.

III. Literature Review

The first task of the Job Recommendation system is to extract the text. In today's world, graphic resumes are the new thing and most of the times the end file is in an image type rather than text. Detecting text in such scenarios will be difficult.

A. Highlighting of Text

Geetika Mathur[1] has proposed a way of using FAST(Features from Accelerated Segment Test) algorithm for highlighting the text. Here the image is first divided into several different blocks. Then the density of each box is checked. Those with less density as well as those with maximum density are removed. Then connectivity of the groups is checked and the blocks near to each other are stitched together. This method came out to be very fast and could be used for different languages. The end result will be a pdf/image with all text remaining.

B. Text Recognition

A. Coates et al . [2] has discussed about text detection and character recognition using Unsupervised feature learning. Here the model learns the required entities directly from the data as opposed to using text specific models. The result of it was much better from conventional methods. The use of spatial pooling[3] played a big role in getting such accuracy.

J. Weinman et al . [4] has proposed to use a probabilistic model that is aided by a lexicon which helps to bias the recognition of the words that the model already knows. The system follows an n-gram model so that the text can be recognised as a non-lexicon. The system treats the spaces between the characters as a character itself allowing the recognition to be guided by either the lexicon or the n-gram model.

C. Resume Matching

Guo et al . [5] has discussed about using content-based approach for a job recommendation. The system uses a vector space model, making every word vector in an n-dimensional space. The identification of word was done using Bag-Of-Words while feature weighting approach used was TF-IDF. With this, any new word which wasn't present in the training corpora can also be recognised.

Duygu et al . [6] proposed a method where the system word categorises every skill in the form of ontology. Each skill would either be a parent skill or a child skill. Once a child skill is found, the whole parent skill is taken into consideration likewise when a parent skill is found to be matching, the child skills are also considered. This approach even though requires setup is very accurate as well as reliable. The ontology can also be updated very easily without requiring to train any data.

Yi et al . [7] approach for this problem is Structured Relevance Models (SRM). These models particularly work on semi structured documents like a resume. This is based on the idea that plausible values for a given field could be inferred from the context provided by the other fields in the record. For SRM we first initiate each field of a given job as a query against the corresponding field of the semi-structured job collection. This is later merged with appropriate weights.Out of the output list, the top rank jobs are only selected. For instance, if two jobs have "Web developer" in the Job Title fields, it is likely that appropriate candidates' resumes for both jobs should have "HTML" or "CSS" in the Resume fields.

Kum Yu et al . [8] proposed a method by using cascaded Information Extraction framework to get all the information from the resumes in a detailed manner. Then the model uses two stages, In the first stage, The resume is segmented into multiple consecutive blocks using Hidden Markov Modeling (HMM) model. This gives the flexibility of customizing of the extent to which the block can have the data. Once a block is defined, A Support Vector Machine model is used to extract and classify the information in a block.

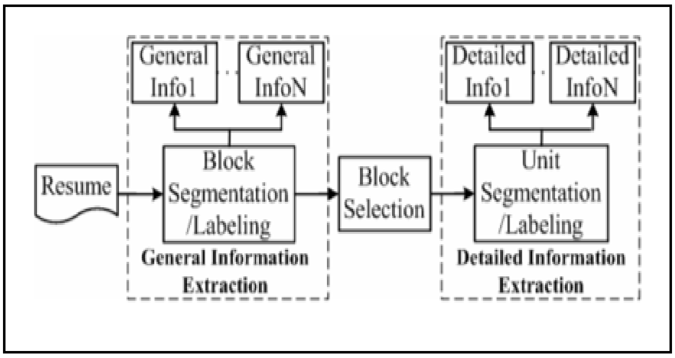

Fig. 1 *Block diagram explanation of an SVM*

Guo et al . [9] discussed about using a similarity matrix to find the most favoured skill. This is achieved by creating a finite state transducer library which can match patterns in the sentence, and extracts related information. Even though with only this a similarity matrix can be formed, Guo discovered that creating a domain-specific ontology increases the accuracy. NDCG approach was used to check for the accuracy.

Rafter et al. began to use Automated Collaborative Filtering (ACF) in their Job Recommender System, "CASPER" [10]. In this kind of system, User profiles are extracted from server logs, that includes: revisit data, read time data, and activity data. All these factors are treated as of high importance among users. The system recommends jobs in two steps: First, the system finds users that match the target specification; second, the jobs that similar users liked/selected will be recommended to the target user. The system uses a cluster-based collaborative filtering strategy. The similarity between users is based on how many jobs they both reviewed, or applied.

Zhu et al .[11] proposed to use Word2Vec, Word2doc and WMD to generate similarities between different skill. The neural network for training uses the Skip-gram model, which aims to find vector representations for each word that are useful for predicting surrounding or near-by words in a sentence. When trained on the right corpora, the end result achieved was exactly the way required. This method can also be coupled with [9] to improve the overall performance. This can be very helpful for conditions when we are trying to resolve the vocabulary difference between a resume and a Job description. For example, If a person knows Android well and has written Android in the resume while the job description asks for Java, this Word2Vec model will be able to show that as the similarity between them is more, A person with experience in android can be matched with a Java requirement.

Kalva et al . [12] used Named Entity Recognition(NER) for finding named entities of the text. Apache openNLP framework was chosen because of great api, community support and free and open source software. The model even though was trained for basic criterias but to be trained also for the skills in a resume. This was achieved by web scraping 3000 job descriptions all around. This approach worked efficiently for matching student's resume to a job description.

Schmitt et al . [13] proposed a method he named MAJORE(Matching Jobs and Resumes), involving two modes respectively referred to as collaborative filtering (Cf) and cold-start (Cs) modes. MAJORE comprises of three modules. The first module computes standard vectorial representations, mapping each recruiter/seeker. The second module builds metrics, either directly, or using a matrix. The third module, implemented as a neural net, trains a specific representation for the job matching task. This is the core of the MAJORE architecture.

Li et al . [14] insists about the analogies that focus on mainly two branches of predictive signals: profile context matching and career path mining and propose a contextual LSTM(Long Short-Term Memory) model. This simultaneously collects signals from both sources by jointly learning latent representations for different types of entities (e.g., employees, skills, companies) that show up in different sources. In particular, it generates the contextual representation by aggregating all the profile information and explores the dependencies in the career paths through the LSTM networks.

Singh et al . [15] uses a method of Recurrent Neural Networks for Job Matching as well. However, while [14] was using LSTm networks stand alone, Singh is using Bi-directional RNNs. Firstly, the model was tested for LSTM and GRU(Gated Recurrent Unit). After testing both the models, it was found out that GRU performs slightly better than LSTM networks. Bi-direction RNN (BiRNN) consists of forward and backward RNN structure (GRU cell). In the forward RNN, the input sequence goes from the first word to the last word, and the model calculates a sequence of forward hidden states. The backward RNN takes the input sequence in reverse order, resulting in a sequence of backward hidden states. To compute the final prediction, we average the output from RNNs in both direction and then apply linear transformation to generate the input to the softmax prediction unit. When tested for data Bi-RNN out-performed uni-directional RNNs.

Smirnova et al . [16] proposes a method to improve the current RNNs. A new class of Contextual Recurrent Neural Networks(CRNN) for recommendation have been proposed that can take into consideration the contextual information both in the input and output layers.This even though is in general terms can be used for job recommendation. A CRNN can learn from the actions of a user about his/her interest and the type of job he/she likes. With this, we can further show the user similar job descriptions so as to make sure that the users goes through maximum number of job opening of his/her interest.

References

[1] Geetika Mathur, Ms. Suneetha Rikhari."Text Detection in Document Images: Highlight on using FAST algorithm", *International Journal of Advanced Engineering Research and Science*,vol.4,no. 3, pp.275-284,2017.

[2] A. Coates *et al*., "Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning," *2011 International Conference on Document Analysis and Recognition*, Beijing, 2011, pp. 440-445.

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, pp. 541–551, 1989.

[4] J. Weinman, Jerod & Learned-Miller, Erik & Hanson, Allen. (2008). A Discriminative Semi-Markov Model for Robust Scene Text Recognition. IEEE, Proc. Intl. Conf. on Pattern Recognition

[5] Guo, X., Jerbi, H., & O'Mahony, M.P. (2014). An Analysis Framework for Content-based Job Recommendation. *ICCBR 2014*.

[6] Celik Ertuğrul, Duygu & Karakas, Askin & Bal, Gulsen & Gultunca, Cem & Elçi, Atilla & Buluz, Basak & Can Alevli, Murat. (2013). Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs.

[7] Yi, Xing & Allan, James & Croft, W. (2007). Matching resumes and jobs based on relevance models. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 809– 810. ACM, 2007.

[8] Yu, Kun & Guan, Gang & Zhou, Ming. (2005). Resume Information Extraction with Cascaded Hybrid Model. ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference.

[9] Shiqiang Guo, Folami Alamudun, Tracy Hammond, RésuMatcher: A personalized résumé-job matching system, Expert Systems with Applications, Volume 60, 2016, Pages 169-182

[10] Rachael Rafter, Keith Bradley, and Barry Smyth. Personalised retrieval for online recruitment services. In Proceedings of the 22nd Annual Colloquium on Information Retrieval. 2000., 2000.

[11] Zhu, Yun & Javed, Faizan & Ozturk, Ozgur. (2016). Semantic Similarity Strategies for Job Title Classification. arXiv:1609.06268

[12] Kalva, Thimma Reddy, "Skill Finder: Automated Job-Resume Matching System" (2013). *All Graduate Plan B and other Reports*. 343.

[13] Thomas Schmitt, Phillipe Caillou and Michèle Sebag. Matching Jobs and Resumes: a Deep Collaborative Filtering Task. GCAI 2016. 2nd Global Conference on Artificial Intelligence. 124-137

[14] Li, Liangyue & Jing, How & Tong, Hanghang & Yang, Jaewon & He, Qi & Chen, Bee-Chung. (2017). NEMO: Next Career Move Prediction with Contextual Embedding.

[15] Liu, D.Z.; Singh, G. A Recurrent Neural Network Based Recommendation System; Available online: http://cs224d.stanford.edu/reports/LiuSingh.pdf

[16] Elena Smirnova, Flavian Vasile "Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks" (2016). arXiv:1706.07684