V.S.B.ENGINEERING COLLEGE, KARUR

Department of Electronics and Communication Engineering

IBM NALAIYA THIRAN

LITERATURE SURVEY

TITLE : Skill Job Recommender

DOMAIN NAME : Cloud Application Development

LEADER NAME : H.Vengateswaran

TEAM MEMBER NAME: M.Saran

S.Sasikumar

R.Sathishkumar

MENTOR NAME : Valli Suseela R

ABSTRACT

This paper presents a job recommender system to match resumes to job descriptions (JD),both of which are non-standard and unstructured/semi-structured in form. First, the paper proposes a combination of natural language processing (NLP) techniques for the task of skill extraction. The performance of the combined techniques on an industrial scale dataset yielded a precision and recall of 0.78 and 0.88 respectively. The paper then introduces the concept of extracting implicit skills – the skills which are not explicitly mentioned in a JD but may be implicit in the context of geography, industry or role. To mine and infer implicit skills for a JD, we find the other JDs similar to this JD. This similarity match is done in the semantic space. A Doc2Vec model is trained on 1.1 Million JDs covering several domains crawled from the web, and all the JDs are projected onto this semantic space. The skills absent in the JD but present in similar JDs are obtained, and the obtained skills are weighted using several techniques to obtain the set of final implicit skills. Finally, several similarity measures are explored to match the skills extracted from a candidate's resume to explicit and implicit skills of JDs. Empirical results for matching resumes and JDs demonstrate that the proposed approach gives a mean reciprocal rank of 0.88, an improvement of 29.4% when compared to the performance of a baseline method that uses only explicit skills.

INTRODUCTION

Formal job search and application typically involves matching one's profile or curriculum vitae (CV) with the available job descriptions (JD), and then applying for those job opportunities whose JDs are the closest match to one's CV, and also considering his/her needs, constraints, and aspirations.A few of the things that a person may consider while doing this matching are: a) required skills mentioned in the JDs and skills possessed by self, b) current salary versus salary offered in the new job, c) future prospects after joining the new job, etc. Some of the entities are easy to extract from a JD, for example, the salary offered in a job. However, some other entities, for example, skill extraction (are Python and Java an animal and an island in Indonesia, respectively, or two object-oriented programming languages) and future prospects of a company (it is subjective as well as dependent upon market conditions), need serious consideration.Though tremendous progress has been made in general purpose search engines, job search engines have made only

LITERATURE SURVEY

A candidate acquires skills through formal education, vocation, internships, and/or previous jobs' experience. In due course of time, the candidate may start identifying (new) relevant jobs based on the basis of these acquired skills. The key function of a job search engine is to help the candidate by recommending those jobs which are the closest match to the candidate's existing skill set. This recommendation can be provided by matching skills of the candidate with the skills mentioned in the available JDs. A common approach while doing a skill match is to use standard keyword matching or information retrieval framework as explained in Salton and Buckley (1988). A few challenges of this kind of approaches are: a) The skill may be mentioned in different forms or in terms of synonyms (e.g. cplusplus, c++; programming, scripting, etc.) in CVs and JDs, b) There could be skills that may not be specified in a candidate's profile or a JD, but can be easily determined by business knowledge (for example, 'java' being an object-oriented programming (OOP) language, its experience also indicates experience of OOP), and c) A skill could be an out of dictionary skill, that is, a not-so-common skill-term missing in the dictionary or from a new unseen domain for which the system may not have skills. A framework for skill extraction and normalization was proposed in Zhao et al. (2015). In this paper, a taxonomy of skill was built and Wikipedia was utilized for skill normalization. In Kivimaki et al. (2013), authors proposed a system for skill extraction from documents primarily targeting towards hiring and capacity management in an organization. The system first computes similarities between an input document and the texts of Wikipedia pages and then uses a biased, hub-avoiding version of the Spreading Activation algorithm on the Wikipedia graph to associate the input document with skills. Colucci et al. (2003) introduced the concept of implicit skills.

Inspired by their work we have explored a new method in this paper to mine implicit skills using word and document embeddings. In Lau and Sure (2002), authors described a methodology for application-driven development of ontologies, with a sample instantiation of the methodology for skills ontology development. In Bastian et al. (2014),the team at LinkedIn built a large-scale topic extraction pipeline that included constructing a folksonomy of skills and expertise and implementing an inference and recommender system for skills. The main idea of a job recommendation system is to provide a set of (job) recommendations in response to a user's current profile. In these systems, the users typically can upload their skills or resume or their job search criterion; similarly, the employers or their agents can upload JDs or skills set needed etc along with information such as location, position and other job specific details.

We mined the web to extract a heterogeneous mixture of JDs from various open-source websites. The entire dataset consists of 1.1 Million mined JDs. It has a substantial mix from multiple domains like IT/Software, Health-care, Recruiting, Education and 48 other such domains. This data is used to train our Word2Vec and Doc2Vec models which are explained further in Section 4. Since no standard large open source dataset exists for the task of CV to JD matching, we approached a research team (Maheshwary and Misra (2018)) who had worked on this problem using deep Siamese Network. The dataset borrowed from them consists of 1314 resumes which came in as a part of summer research intern application at their company and a set of 3809 JDs from various domains. We have used this dataset for our full job recommender system evaluation so that we can compare our results with some existing published results.

REFERENCES

[1] Al-Otaibi, S. T., and Ykhlef, M. 2012. A survey of jobrecommender systems. International Journal of Physical Sciences 7(29):5127–5142.

[2]Paparrizos, I. K.; Cambazoglu, B. B.; and Gionis, A. 2011. Machine learned job recommendation. In RecSys.

[3] Lau, T., and Sure, Y. 2002. Introducing ontology-based skills management at a large insurance company. In In Proceedings of the Modellierung 2002, 123–134.

[4] Maheshwary, S., and Misra, H. 2018. Matching resumes to jobs via deep siamese network. In Companion of the The Web Conference 2018 on The Web Conference 2018, 87 88. International World Wide Web Conferences Steering Committee.