Towards Entity Recognition and Relation Extraction from Job Description Text

Shraddha Mukesh Makwana
University of Alberta
CCID:smakwana
smakwana@ualberta.ca

Pranjal Dilip Naringrekar
University of Alberta
CCID:naringre
naringre@ualberta.ca

1 Task

Finding the perfect match of job to candidate's profile without trivial recommendation is difficult and not a straightforward task. How to successfully fill in the job vacancy heavily depends upon the skill-set mentioned in the job description and resume of the candidate. Least skill-gap between these two factors ameliorates the perfect fit between job-seeker's profile and job position. Moreover, considering competitiveness and dynamic requirements within the market, is giving rise to more skill-based matching of resumes for jobs. Traditional job search systems perform simple data-mining based on keyword similarity, and do not take into account the interlinks between entities (Senthil Kumaran and Sankar, 2013). Therefore, towards the goal of developing more efficient job recommendation and skill discovery, our current task will focus on extracting the entities and relations from the job descriptions, which are basic building blocks for constructing the job search knowledge graph.

Hence, in this task we try to perform joint Named Entity Recognition and Relation Extraction between the entities to build a better match between job and candidate's profile skills. Our aim is to build the relation extraction model which will be a classifier that predicts a relation 'r' for a given pair of entities e1, e2. For an instance, a candidate might have extensive skills in "Java" programming, but the job description of interest requires knowledge in "J2EE" framework, which is essentially based on "Java"; hence relation extraction will make sure it understands that "Java" is entangled with "J2EE" and perform better resume shortlisting.

Figure 1, depicts the basic architecture of the task, wherein the job description dataset will be passed to the relation extraction classifier and the output of the same will be relation triples from the input text material. As an example, if the job description contains entities like "Bachelor" and "Computer Science" and their relationship is "degree_in", in that case the output relation triples will be ("Bachelor", "degree_in", "Computer Science").

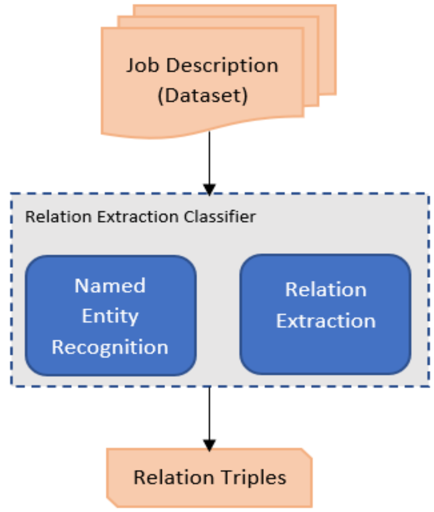

Figure 1: System Architecture

Therefore, solving this task of relation extraction from job description will help to address upon the foundation for a tool that finds field-relevant skills listed on the resume and to uncover which job skill it is more pertinent to. It's application is majorly focused on transferring the job searching task from keyword searching to candidate model matching (Guo et al., 2016).

2 State-of-the-art

The task of Entity Recognition and Relation Extraction have been previously solved using various approaches. The architecture proposed by de Groot et al. (2021) studies link prediction methods for quantifying the relatedness between skills and occupations using Node2Vec. The performance of their model exceeded two different link prediction methods, which were based on preferential attachment (PA). Biancofiore et al. (2021) extracted stronger relations by merging some sub-graphs from state-of-the-art solutions like Dbpedia and new triples scraped from external sources. Wang et al. (2021) propose a keyword-based search engine for recruitment and staffing by extracting relation triples using transformer-based methods. They used BERT for implementation, which performed better in finding good candidates than Linx.

We plan to achieve high performance for this task by exploring multiple state-of-the-art Natural Language processing based techniques such as Markov Logic Networks proposed by Niu et al. (2012) and experimenting with other different types of feature extraction to understand and evaluate Relation Extraction triples. We will also try HD-SKG solution done in the paper by Zhao et al. (2017), which performs transformer-based models that worked particularly well for this task. We will not limit our work to the above mentioned task and will try to experiment with many more aspects like fine-tune a BERT model for NER and Relation Extraction using spaCy3.

For assessing the performance of our work, we will be using the same evaluation metrics used by Zhao et al. (2017), which are precision, recall and f-measure. The F1-score (0.28) for relation extraction by Angeli et al. (2015) who proposed the openIE tool will remain our baseline. Our aim would be to maximize this F1-score value which we calculate using the predictions of the models. In this way, we will try to achieve the best-performing model for the task.

3 Available Data

We will be using the dataset called “Trainrev1” which was provided over the public platform Kaggle. It was created by collecting job descriptions related to software engineering, hardware engineering, and research from various companies’ websites and job portals. The data was then stored in a csv file with about 2,44,769 distinct job descriptions, with fields like salary range and job location given in it’s raw and normalized form.

We plan to partition the dataset into 3 subsets, the Train Dataset with about 65% records, the Dev Dataset with about 15% records, and the Test Dataset with 20% of the records.

If we look at any one record of the dataset as an example, we will be able to see 10 fields in the data as described in Table 1.

Field	Value
Id	12612628
Title	Systems Analyst
JobDescription	Our clients are looking for...
LocationRaw	Dorking, Surrey
LocationNormalized	Dorking
ContractTime	permanent
Company	Martin International
SalaryRaw	20000 - 30000/annum
SalaryNormalized	25000
SourceName	cv-library.co.uk

Table 1: Sample Data Record

4 Available code

We found a code by Zhao et al. (2017) with the model that performs Relation Extractions for domain specific knowledge graph construction. After execution of this code, we found that it gives an F1-score score of 0.6 on the test set.

https://github.com/Rvlis/Implementation-of-HDSKG-using-BERT

Along with this, we attempted to execute a code by Grainger et al. (2016) which introduces Semantic Knowledge Graph and leverages an inverted index, for determining relationships and to represent nodes (terms) and edges (the documents within intersecting postings lists for multiple terms/nodes). Their framework exposes a RESTful API, and in order to run their code in an easiest way, one simply needs to send a corpus of documents to the Semantic Knowledge Graph API. However, due to older versions we were not able to successfully replicate their results. The code can be found at:

https://github.com/careerbuilder/semantic-knowledge-graph

5 Repository URL

The url for our project repository can be found at:

https://github.com/shr1911/cmput656-job-knowledge-graph

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Giovanni Maria Biancofiore, Tommaso Di Noia, Eugenio Di Sciascio, Fedelucio Narducci, and Paolo Pastore. 2021. Guapp: Enhancing job recommendations with knowledge graphs. *11th Italian Information Retrieval Workshop*.

Trey Grainger, Khalifeh AlJadda, Mohammed Korayem, and Andries Smith. 2016. The semantic knowledge graph: A compact, auto-generated model for real-time traversal and ranking of any relationship within a domain. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 420–429. IEEE.

Maurits de Groot, Jelle Schutte, and David Graus. 2021. Job posting-enriched knowledge graph for skills-based matching. *arXiv preprint arXiv:2109.02554*.

Shiqiang Guo, Folami Alamudun, and Tracy Hammond. 2016. Résumatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, 60:169–182.

Feng Niu, Che Zhang, Christopher Ré, and Jude W Shavlik. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28.

V Senthil Kumaran and A Sankar. 2013. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (expert). *International Journal of Metadata, Semantics and Ontologies*, 8(1):56–64.

Yan Wang, Yacine Allouache, and Christian Joubert. 2021. Analysing cv corpus for finding suitable candidates using knowledge graph and bert. In *DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications*.

Xuejiao Zhao, Zhenchang Xing, Muhammad Ashad Kabir, Naoya Sawada, Jing Li, and Shang-Wei Lin. 2017. Hdskg: Harvesting domain specific knowledge graph from content of webpages. In *2017 ieee 24th international conference on software analysis, evolution and reengineering (saner)*, pages 56–67. IEEE.