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 Literature Review

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). 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.

Biancofiore et al. (2021) proposed GUapp which is a tool for searching and recommending job openings in the Italian public sector. The platform provides recommendation services with the goal of matching user skills and requests with job openings in a specific time frame. They extracted stronger relations by merging some sub-graphs from state-of-the-art solutions like Dbpedia and new triples scraped from external sources.

A method for rating applicants based on keywords related to competence was proposed by Wang et al. (2021). To begin with, they determined the score of matching between a competency keyword and a corpus of CVs using the TF-IDF Vectorizer. Second, based on two types of competency keywords they used the Weighted Average Method to create a global CV score. Third, they constructed a Knowledge Graph (KG) from a structured Competence Map that can classify bidirectional association. Finally, they proposed using BERT's Named-Entity Recognition to better identify tokens in the client's input questions and tested it on CVs from the Human Resource Department.

2 Draft of our Methods

We aim to solve the problem by using Named Entity Recognizer (NER) and transformer based model (BERT) to extract entities and relationships, respectively and store the constructed KG in Neo4j.

For our implementation, we have divided the dataset as 15% dev data, 70% as train data and 15% as test data. We utilise the UBIAI text annotation tool to collect training data because of its flexible interface, which allows us to quickly switch between entity and relation annotation. We will then convert our annotated data to a binary spacy file before we can train the model. We separate the UBIAI annotations into train/dev/test and save them separately. Following this, we would firstly extract entities using the pre-trained fine-tuned NER model to find skills, diploma, diploma major and years of experience. Secondly, we would implement the HDSKG solution done in the paper by Zhao et al. (2017), which uses BERT model to extract the relationships. Lastly, we will evaluate our method on 15% test data. The F1-score (0.28) for relation extraction by Angeli et al. (2015) who proposed the openIE tool will remain our baseline and our aim would be to maximize this F1-score value.

We will not limit ourself to beforehand mentioned work but will also try 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.

3 Draft of the Evaluation Protocol

For assessing the performance of our work, we will be using the same evaluation metrics used by Zhao et al. (2017), which is F1 score. It's calculation is the harmonic mean of Precision and Recall, as

No.	Task	Start Date	End Date	Status
1	Study of Problem Statement	27th Jan	31st Jan	Done
2	Literature Review	1st Feb	6th Feb	Done
3	Study and Execute Open-Source Code	7th Feb	14th Feb	Done
4	Plan Implementation Methodology	15th Feb	19th Feb	Done
5	Implementation	20th Feb	7th Mar	In-Progress
6	Visualization and Evaluation	8th Mar	14th Mar	In-Progress
7	Buffer For Implementing Blockers (If Any)	15th Mar	19th Mar	To-Do
8	Final Report Draft and Review	20th Mar	27th Mar	To-Do
9	Improvement of Final Report based on Feedback	28th Mar	5th Apr	To-Do
10	Final Report Submission	6th Apr	8th Apr	To-Do

Table 1: Plan for Completion of the Project.

follows:

$$F1 = 2 \times (\frac{(precision \times recall)}{(precision + recall)})$$

The evaluation measure is calculated based on the overlap between the predicted and actual extracted relationships. Our evaluation function will give credit to partial matches between gold and predicted relationships. The partial credit is proportional to the intersection of the two relationships, and it is normalized by the length of the two entities. The gold label standards for this task will be handcrafted by us.

4 Draft of Results

We started our work with the data gathering process. We successfully built the input dataset from the UBIAI text annotation tool wherein a sample sentence would look like: 'Text=["'2+ years of non-internship professional software development"']'. Along with this, we also converted this annotated data to binary spacy file.

Also, for better performance we tried implementing the Dictionary-based named entity recognition and Bi-LSTM-CRF model to extract entities. Dictionary-based named entity recognition categorized datasets based on collected dictionaries or user-defined dictionaries (Neelakantan and Collins, 2014). However, there is a disadvantage of having to manually organize dictionaries, and because it is necessary to deal with constantly changing and emerging new words over time we tried the Bi-LSTM-CRF model which showed meaningful performance in time series data, using supervised learning-based word embedding and non-supervised learning-based word embedding from a large corpus (Huang et al., 2015).

We are currently implementing the pre-trained NER model defined by Wild et al. (2021). Also, we started working on the code that implemented the BERT model provided by Yang et al. (2020). However, it has dependencies on lower version of python libraries.

5 Plan for Completing the Project

We plan to complete our project task according to the specifications mentioned in Table 1. Along with this, the implementation life-cycle will be more into the incremental phase similar to Agile Methodology, delivering end-to-end features in each phase.

6 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*.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Arvind Neelakantan and Michael Collins. 2014. Learning dictionaries for named entity recognition using minimal supervision. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 452–461, Gothenburg, Sweden. Association for Computational Linguistics.

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.

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*.

Simon Wild, Soyhan Parlar, Thomas Hanne, and Rolf Dornberger. 2021. Naïve bayes and named entity recognition for requirements mining in job postings. In *2021 3rd International Conference on Natural Language Processing (ICNLP)*, pages 155–161.

SungMin Yang, SoYeop Yoo, and OkRan Jeong. 2020. Denert-kg: Named entity and relation extraction model using dqn, knowledge graph, and bert. *Applied Sciences*, 10:6429.

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.