Rapid Graph Generation from Job Descriptions: combining Taxonomies and LLMs

Kaan Karakeben, Henri Egle Sorotos, Alisa Milchevskaya, and Ahmad Assaf

Beamery, London UK
kaan.karakeben@beamery.com, henri.egle-sorotos@beamery.com,
alisa.milchevskaya@beamery.com, ahmad@beamery.com
www.beamery.com

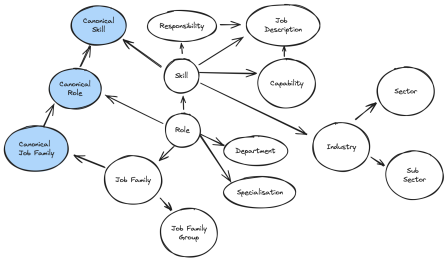
Abstract. Job Descriptions are a common currency for hiring talent. Whilst they are ubiquitous, they are also unstructured, difficult to analyse, and have no common format. In this paper, we present a novel approach to generate knowledge graphs from job descriptions by utilising a bespoke Large Language Model to extract and reconcile concepts to our ground truth taxonomy of HR entities.

Keywords: knowledge graph · unstructured data · taxonomies · LLM.

1 Introduction

Human Resources (HR) is a core business function of almost all organisations. They manage the most important asset of many companies, its people. HR is a data-rich, but consistently data-poor function. This is not a trivial problem to solve because the causes are multi-faceted. HR data is unstructured, and often qualitative. Artefacts include performance reviews, resumes, and job descriptions (JDs). This data needs to be structured before being used as input for AI, analytics, and reporting.

Fig. 1. Beamery's HR Domain Model



Another problem is that HR data is often siloed. Systems are often not integrated, and data is often not shared across the organisation. To confront the

intricate challenge of how data is interconnected, we have developed a graph domain model [4] to represent concepts found in the HR domain. This is represented in Resource Description Framework (RDF) format, and is conceptually shown in Figure 1.

Crucially, it provides a common language of canonical taxonomies for HR data, and can be used to reconcile data across different systems. This domain model can be used to structure and reconcile data from HR artefacts, such as JDs. Notice that skills are the central concept in the domain model, and are related to other concepts such as qualifications, required capabilities, and responsibilities. JDs are a key source of this data and are found in companies globally. The paper will discuss our approach using an in-house fine-tuned Large Language Model (LLM) to extract these domain concepts from JDs, and reconcile them to our domain model.

2 Approach

Our approach to generating a knowledge graph from job descriptions is a two-step process:

1. **Concept Extraction and Reconciliation:** We use a bespoke LLM to extract concepts from job descriptions and reconcile to our canonical domain model.
2. **Graph Generation:** We generate a graph from the extracted raw concepts with relationships to canonical skills in our domain model.

2.1 Concept Extraction and Reconciliation

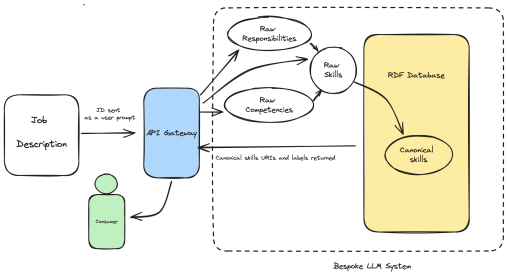
Job vacancy datasets are now abundantly available across various platforms. They provide insights into the dynamic demands of the labour market and enhance job matching processes [1]. A JD is a document containing unstructured and semi-structured natural language. This is a mix of free text and lists. There is no common format for JDs, and they can vary widely in length and content. They contain wealth of information about the skills, qualifications, competencies and prior experience required for a role.

The advancement of generative LLMs, particularly those that are instruction-tuned, has led to their widespread application across a broad spectrum of fields [2]. These applications frequently achieve highly competitive outcomes, factually demonstrated by OpenAI's GPT-4 upon release in 2023 [3]. Our fine-tuned LLM was trained on Beamery's talent graph [4] which contains proprietary data from a variety of HR artefacts. These concepts are found in our domain model, and the LLM is built to extract only these specific concepts from JDs. Prompt Engineering [5] is used to prime the bespoke system to act as a highly experienced HR professional with experience extracting key terms from JDs. The task is shown to the model as a system prompt via few-shot learning [6], which prepares the LLM to extract concepts from the JD. This enables in-context learning of the

system to successfully extract the following concepts 1) skills, 2) capabilities, and 3) responsibilities. Our work builds upon that by Zhang et al. [7]. From the perspective of the system, the JD is a prompt, and the concepts are the completion.

We assume skills are the most granular and important concept in model. Capabilities and responsibilities are the sum of the skills required for a role, as per the domain model in Figure 1. Because of this, we send the extracted capability and responsibility concepts to the system again, and ask it to extract skills from these concepts. This creates a complete graph of skills, capabilities and responsibilities.

Fig. 2. Bespoke LLM System Diagram



2.2 Graph Generation

Underlying the bespoke LLM is a taxonomy of "canonical skills" that are designed to be mutually exclusive and collectively exhaustive (MECE). These are stored using the Simple Knowledge Organisation System (SKOS) [9] in graph. The canonical taxonomies are the common language understood by our AI and downstream applications.

What makes this approach unique is that the LLM returns extracted concepts as concepts in our canonical domain model of specific skills. These terms returned have associated graph URIs. Whilst extracting concepts using a generic LLM is a useful tool, it would not reconcile to our canonical skills in the domain model. Generic LLMs result in a high degree of cardinality of concepts extracted. For instance, any vanilla LLM could extract "Python" and "Python programming" as separate skill concepts. Previously seen and reconciled concepts are used to train and improve the LLM system and are persisted in the graph.

Edges between the raw skills concepts and the canonical skills are created and persisted with edges to common URIs. This allows us to generate a graph of the original JD, extracted concepts, and their relationships to the canonical skills. Each group of concepts extracted from a JD is stored as a named graph

using both graph per source and graph per resource [8]. Provenance is a key value add. All JDs are stored in a common format, and linked to canonical skills.

3 Conclusion

JDs are a common currency for hiring talent. As initially stated, they are unstructured, difficult to analyse, and have no common format. The method in this paper allows us to generate a graph from JDs, and have a common currency for applications. It has been shown that by linking these concepts to a canonical taxonomy, we can reconcile data across different systems and feed data into AI, analytics and reporting. The benefits of linked data can be realised, such as SPARQL queries. At Beamery, this approach is in production and used to power AI applications such as matching and ranking candidates to jobs [12].

The method proposed is not without its limitations. Some potential problems include the fact the LLM may not always extract the correct concepts from the JD. We may see hallucinations or incorrect concepts extracted. Additionally, the reconciliation model may not always reconcile the concepts to the correct canonical skills. There may be concepts in the JD that are not in the canonical skill taxonomy. Despite this, the method proposed is significantly more efficient than manual methods. It allows results to power AI, analytics and reporting with JD data. In future, a similar mechanism could be used to link JD data to existing skills data such as ESCO [10] or ONET [11]. Additionally, there is the potential to see new insights from the data, and to power new applications that were not possible before.

References

1. K. Balog, Y. Fang, M. De Rijke, P. Serdyukov, and L. Si, "Expertise retrieval," *Foundations and Trends in Information Retrieval*, vol. 6, no. 2-3, pp. 127-256, 2012.
2. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., "Training language models to follow instructions with human feedback," *arXiv preprint*, 2022. arXiv:2203.02155.
3. OpenAI, "GPT-4 technical report," *arXiv preprint*, 2023. arXiv:2303.08774.
4. "Beamery," 2022. Retrieved from https://beamery.com/resources/blogs/knowledge-graphs-the-future-of-talent-management.
5. E. Saravia, "Prompt Engineering Guide," 2022. Retrieved from https://github.com/dair-ai/Prompt-EngineeringGuide.
6. T. Brown et al., "Language Models are Few-Shot Learners," *arXiv preprint*, 2020. arXiv:2005.14165v4.
7. M. Zhang, K. N. Jensen, S. D. Sonniks, and B. Plank, "SkillSpan: Hard and Soft Skill Extraction from English Job Postings," *arXiv preprint*, 2022. arXiv:2204.12811.
8. L. Dodds and I. Davis, *Linked Data Patterns: A pattern catalogue for modelling, publishing, and consuming Linked Data*, 2022. Retrieved from https://patterns.dataincubator.org/book/.
9. W3C, "SKOS Simple Knowledge Organization System Reference," 2009. Retrieved from https://www.w3.org/TR/skos-reference/.

10. “ESCO,” 2024. Retrieved from https://ec.europa.eu/esco/portal/home.
11. “O*NET,” 2024. Retrieved from https://www.onetonline.org/.
12. “Beamery AI Talent Match,” 2024. Retrieved from https://support.beamery.com/hc/en-us/articles/9671611196305-Beamery-AI-Explained.