Skill Categorization from Job Descriptions using NER

Tabish Khan
309670657
Syracuse University
tkhan12@syr.edu

Abstract

This paper introduces a novel method for extracting and classifying talents from job descriptions using sophisticated text mining techniques: skill categorization from job descriptions using NER. The study makes use of Named Entity Recognition (NER) in conjunction with an optimised BERT model to precisely identify both general and specific technological and soft skills from an extensive dataset of job descriptions that have been scraped from the internet. The project takes into account the ethical implications of data usage and technology in human resource contexts as it tackles problems with data sampling, annotation, and model optimisation. The results demonstrate the potential of NER in job recommendation systems and provide insightful information about developments in the labour market and skill demands.

Keywords: Named Entity Recognition (NER), BERT, Recommendation System, Skill Analysis, Machine Learning, Text Mining

1. Introduction

In this work, I provide a novel text mining approach designed to identify job abilities from job descriptions: "Categorization from Job Descriptions using NER." Using sophisticated Natural Language Processing methods, especially Named Entity Recognition (NER) and a well-tuned BERT model, SkillSpotter examines how job needs are changing over time. Using a dataset of web-scraped job descriptions, I produced training data using BIO tags to optimise the BERT model.

Further augmenting my approach, I established a comprehensive taxonomy of skills based on the skills explicitly mentioned in the job descriptions and the extensive list of skills from O*NET OnLine. This taxonomy consists of technology skills and soft skills and facilitates the creation of bespoke taxonomies for each job description of interest. Specifically, my study zeroes in on 34 technology-based job titles, selected for their relevance to my research objectives. For each of these job titles, I meticulously maintain a distinct taxonomy, ensuring a tailored and nuanced understanding of the skill requirements.

I outline my approach's methodology in the sections that follow, starting with data transformation, cleaning, and sampling and ending with model training and inference. After that, I assess the effectiveness of my suggested models and go over their advantages, disadvantages, and possible advancements. Lastly, we go over my findings and a few applications for my model. Through my investigation of these cutting-edge methods, I hope to further the continuous progress towards more individualized and efficient recommendation systems.

2. Related Work

A "Skill2Vec" architecture was previously described by Duyet et al. [3]. It uses a special embedding technique to represent skills in a multi-dimensional space, making it easier to identify the links between various abilities. Its main neural network technique is inspired by Word2Vec, with a focus on vector space representation of skills. This method is similar to how I utilise BERT, which leverages NER with an entity identification focus, to extract context-aware skills from job descriptions.

Transformer designs have not been used in the majority of skill extraction advancements in recent years, such as those described in "SkillNER: Mining and Mapping Soft Skills from Any Text" by Silvia Fareri et al. [2]. Instead, techniques comparable to Multi-Layer Perceptrons (MLP). Although MLPs are useful, using BERT as I have done here offers a contextual knowledge of entire phrases as opposed to simply words. This distinction is important because it makes it possible to extract abilities more accurately and nuancedly, especially in complex job descriptions where context is important.

3. Method

As illustrated in Figure 1, the SkillSpotter project's technique uses Named Entity Recognition in an organised manner to mine skills from job descriptions. To guarantee the text's quality and relevancy, a thorough data cleaning step is taken at the start of the process. The next steps, which are necessary to get the text ready for NER, are tokenization and BIO tagging. Application of the Distilbert-base-uncased model, which was chosen especially for its efficiency in NER tasks and lightweight nature, forms the basis of the technique. To guarantee that the model accurately recognises and classifies skills, it is painstakingly trained and assessed.

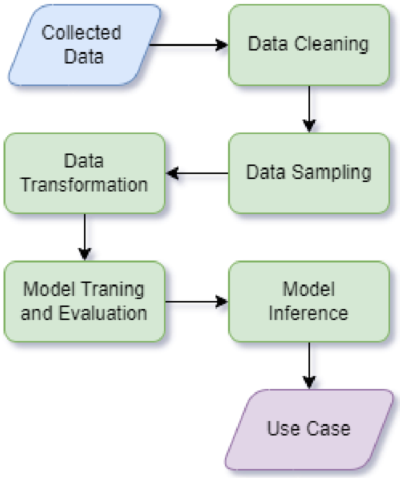


Figure 1. Methodology Flowchart.

3.1. Data Description and Exploratory Data Analysis

180K web-scraped job descriptions from several websites made up the dataset I used. Every job description included a column for talents that were specifically listed in the job description along with a unique Job ID. Many tech and non-tech job titles for positions with US bases were included in the dataset.

I had roughly 87K distinct job titles in the dataset. The majority of these job titles were ambiguous, such as "Team Member" and "Crew Member," and many of them came from the healthcare sector.

I followed the method depicted in Figure 3 for developing the skills taxonomy. I combined O*Net OnLine's technical and soft capabilities into a comprehensive list of skills and incorporated all the special abilities that were specifically listed in the job descriptions.

3.2. Data Cleaning and Sampling

The data cleaning and sampling process was meticulously structured. I used the BeautifulSoup 4 library to parse the HTML. I removed rows with a null ’Job Description’, but retained those where the skills column was null. This was crucial to ensure data integrity and reliability. I also removed some noticeable incorrect annotations like empty quotes and words like ’auto’.

I chose to focus the scope of this project on technology-related jobs and thus manually curated a list of 34 jobs from the 80K unique job titles where I had at least 30 job descriptions for each title.

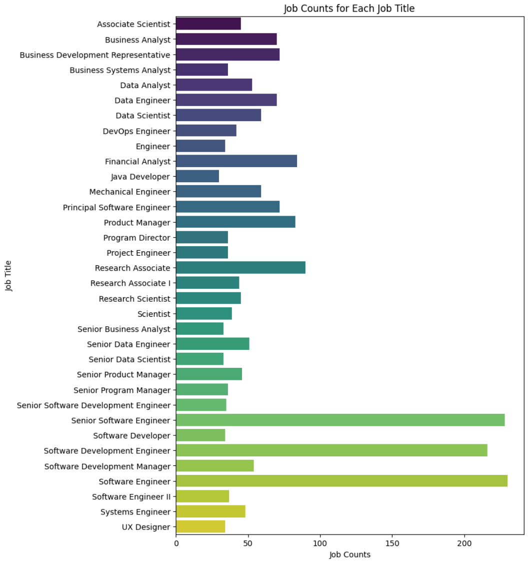


Figure 2. Job counts for each job title.

The data shown in Figure 2 indicates an imbalance, with the majority of job descriptions being for Software Developer and Senior Software Developer positions.

3.3. Data Transformation

The data went through an extensive transformation before the model training process. I started by

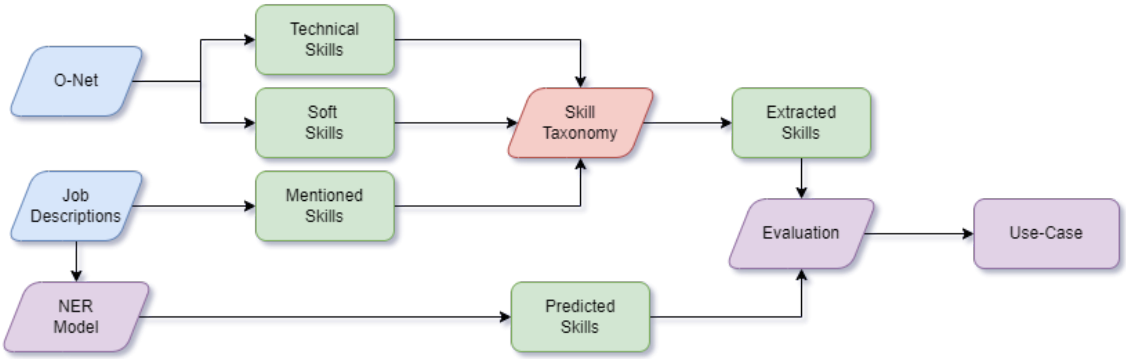


Figure 3. Skills Taxonomy Flow Diagram.

creating a map between the original Job ID and a new, simpler Job ID (one-to-one map), and also mapped each Job ID to the respective Job Title for later analysis.

The data then underwent stop-word removal and a 2 part tokenization process - splitting each job description into sentences and each sentence into words. The 2 part tokenization process was the crucial step for BIO tagging the words.

Job Id	Sentence	Tags
1	[overview, data, integration,. . .	[O, B, I, O...
1	[individual, expected, key,...	[O, O, O,...
1	[position, data, role,...	[O, B, O,...

Table 1. Bio Tagged Sentences.

BIO tagging is the process of assigning classification tags to each word token as shown in Table 1. The three tags represent B - Beginning of the phrase, I - Inside of the phrase, and O - Outside of the phrase. The BIO tags were created using the pattern matching of combinations of tokens in the Job Description and in the skill taxonomy. Finally, I flattened all job descriptions into a dataset of sentences and their corresponding tags and removed all sentences where there were no phrases present.

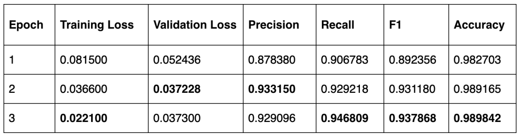
I used the distilbert-base-uncased tokenizer to again tokenize my sentences so that they could be represented in the distilbert-learned word embeddings. All BERT models use sub-word tokenization which splits some words into more tokens. This created an issue with the maximum token length of my model and also led to a misalignment in the BIO tags. Thus I truncated and padded all sentences to 512 tokens and wrote a script to align the tags with the sub-word tokenized sentences.

3.4. Model Training

18K words made up the final converted dataset, which was divided 80:20 into training and testing samples. I modified the distilbert-base-uncased transformer model with my training set of data. I chose Distilbert uncased since it is a lighter and faster version of the full BERT model. Using a compact architecture with 40% fewer parameters, Distilbert uncased has been trained to mimic the probabilities of the full BERT model, allowing it to operate 60% faster while retaining over 95% of BERT's performance [1].

I used a weight decay of 0.01 and a constant learning rate of 0.00002 over my three epochs of training. Every training session lasted forty minutes. The model's greatest F1 score was 93.78%, and its accuracy was 98.98%. Figure 6 shows that as soon as training begins, both the training and validation losses start to decline. After the second epoch, the validation loss fully converges, but the rate of decrease of the training loss slows down.

4. Results



Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	0.081500	0.052436	0.878380	0.906783	0.892356	0.982703
2	0.036600	**0.037228**	**0.933150**	0.929218	0.931180	0.989165
3	**0.022100**	0.037300	0.929096	**0.946809**	**0.937868**	**0.989842**

Table 2. Model Metrics Results.

Table 2 indicates that the model performs exceptionally well in correctly predicting or classifying the data. An accuracy of nearly 99% is particularly

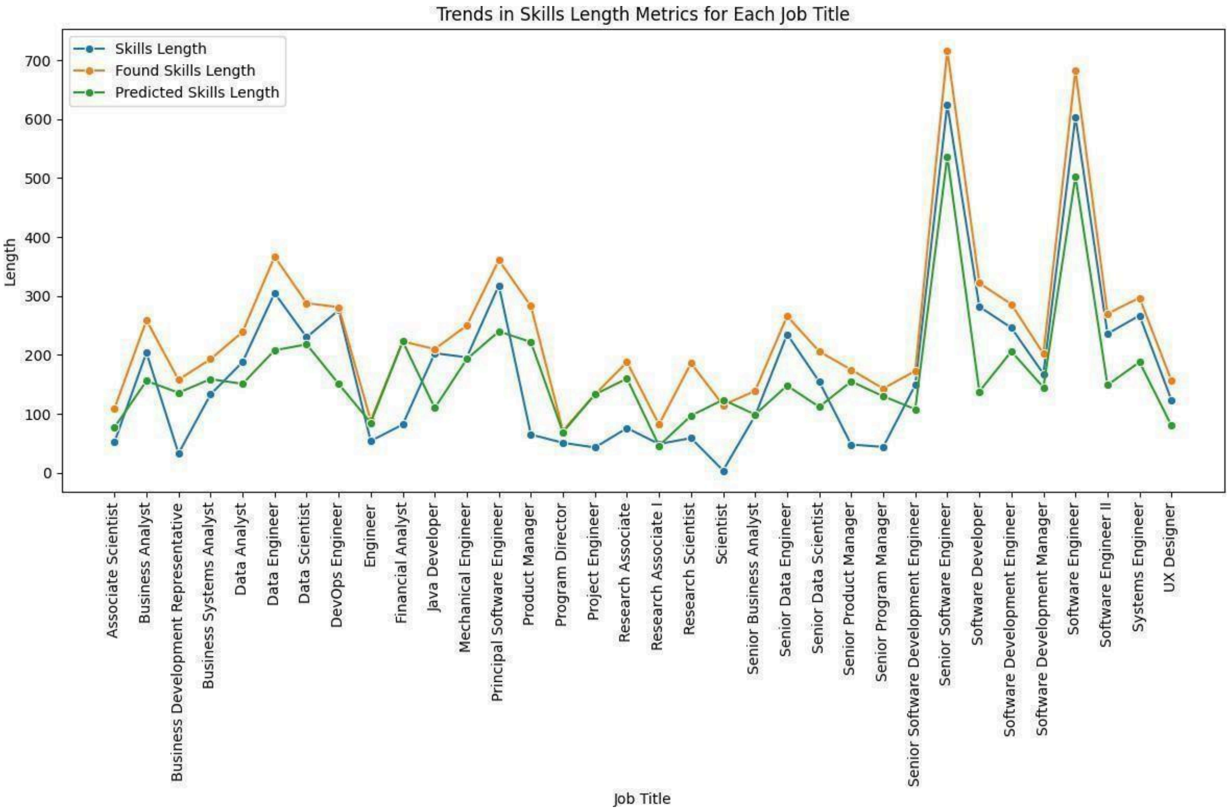


Figure 4. Trends in skill length metrics for each Job Title.

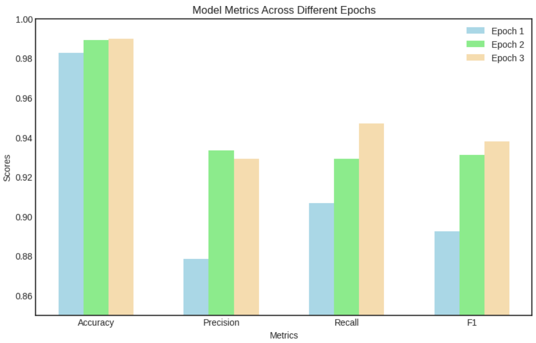


Figure 5. Model Metrics.

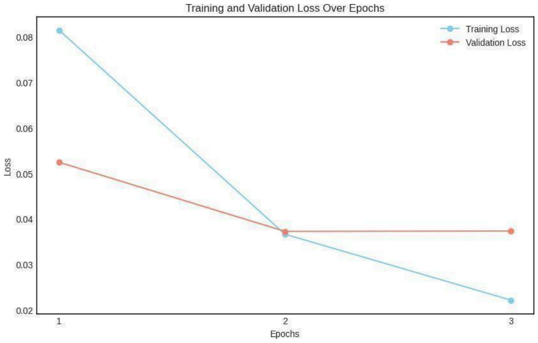


Figure 6. Training and Validation loss variation.

impressive, suggesting that the model has learned the semantic meaning of both soft skills and technology skills.

A high F1 score of 93.78% suggests that precision and recall—the model's capacity to identify all positive samples—are strongly balanced. Precision refers to the model's capacity to identify as positive only those samples that are genuinely positive. This implies that the model's predictions are trustworthy in addition to being accurate.

4.1. Model Inference

I tested the model on my entire dataset of job descriptions and various example sentences generated by ChatGPT on its ability to :

- Detect known skills from the Job Descriptions.
- Detect new skills not present in the skill taxonomy.
- Catch abbreviated skills.
- Catch misspelled skills.

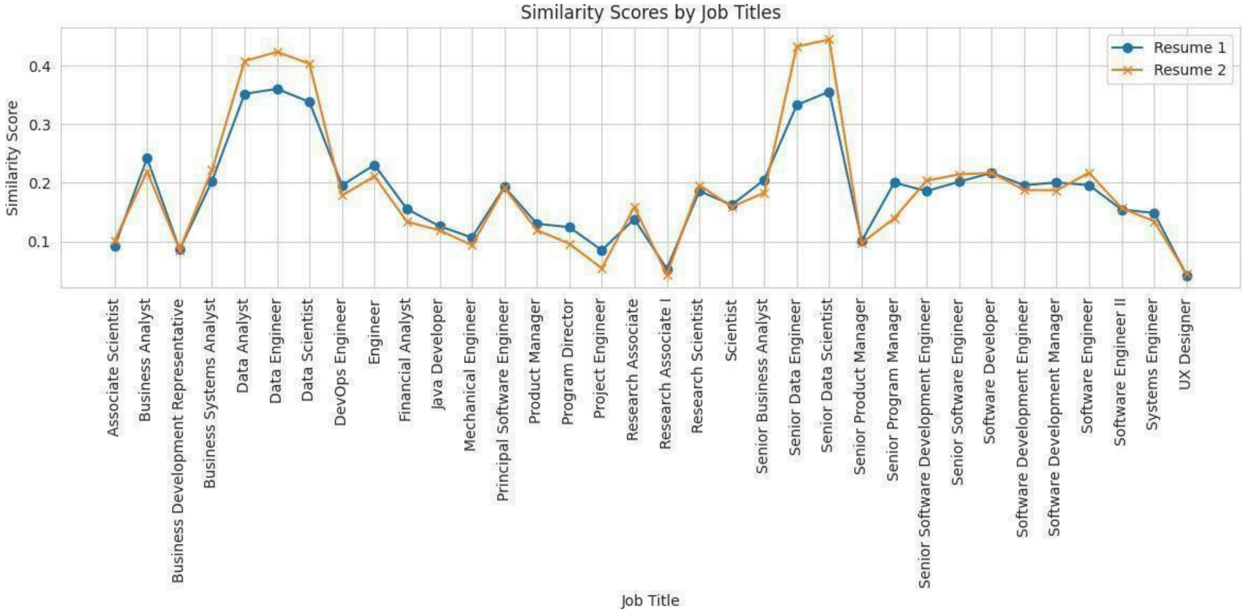


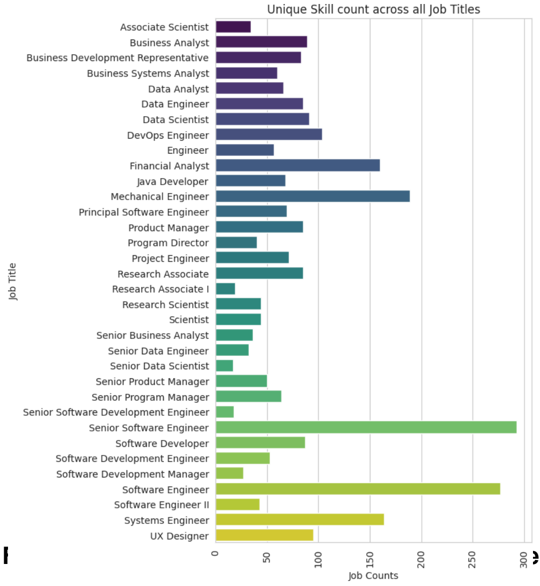
Figure 7. Similarity Score for 2 sample resumes for each Job Title.

Figure 4 shows that I was able to find a lot more skills from the data using pattern matching from my skill taxonomy than were specifically stated in the job posting. With about 99% accuracy, my model was able to replicate the pattern matching results. Although Figure 4 shows that my model did not match the results, it does aggregate for all abilities found for each job title.

My model performed well when I tried it to specifically identify skill words from ChatGPT-generated sentences. However, it had some difficulty when it came to identifying new skill words. Some abbreviations, such as JS for JavaScript, could be recognised by the model, but not ml for machine learning. I spelt a few skill words incorrectly, but the model caught seven out of ten small spelling errors.

In order to find talents that weren't included in other job titles, I created distinct lists of skills for each job title using the model. These abilities are highly valued for all class of job titles.

Using the data from Figure 2, I compared similar jobs to see what makes them unique. When taking a closer look at Jobs like Data Engineer and Senior data engineer, even though they seem similar, the Data Engineer role is centered around technical proficiency in data management, analytics, and development, on the other hand, the Senior Data Engineer position requires a higher level of expertise, encompassing advanced technical skills, strategic leadership, and business acumen, particularly in big data technologies and cloud platforms. This backs up the fact that these unique skills have high feature importance for each job title class



4.2. Use Case

Skills can be extracted from any text that has been tokenized by the Distilbert tokenizer using this paradigm. In Figure 7

I attempted to extract talents from two sample resumes using this model, and I determined the cosine similarity between all the abilities needed for each job title. In addition, this model can be utilised as a recommendation engine, suggesting positions to candidates whose skill sets closely match those listed in the job description.

5. Conclusion

All things considered, the model worked effectively; in roughly 15% of instances, it identified new skills in job descriptions that weren't included in my skill taxonomy. So, it would be ideal to use a flexible skill taxonomy in conjunction with a BERT model to identify every skill included in job descriptions and to update the taxonomy as needed for a recommendation system.

6. Ethics Statements

Ethical Use of Open-Source Data

This data is open source and scraped from the web. I ensured that the use of this data aligns with the terms set by the data providers. The data is being used for this academic project and this project would not be used for commercial purposes.

Over-Reliance on Technology

Sole reliance on the NER models like SkillSpotter for screening candidates and job searching can be problematic. It's essential to have human oversight to interpret and contextualize the model's findings, especially in complex and nuanced fields like human resources.

Data Imbalance

I have trained this model on a dataset which only contained Job Descriptions form companies in the US. I may be missing out on region specific skills if model is used to infer on a dataset with jobs from another geolocation. Similarly, I have trained the model to work with only tech specific roles and it may not work well for other roles..

References

[1] Hadeer Adel et al. "Improving crisis events detection using distilbert with hunger games search algorithm". In: *Mathematics* 10.3 (2022), p. 447.

[2] Silvia Fareri et al. "SkillNER: Mining and mapping soft skills from any text". In: *Expert Systems with Applications* 184 (2021), p. 115544.

[3] Le Van-Duyet, Vo Minh Quan, and Dang Quang An. "Skill2vec: Machine learning approach for determining the relevant skills from job description". In: *arXiv preprint arXiv:1707.09751* (2017).

[4] Xu, M., Jiang, H., & Watcharawittayakul, S. (2017)." ALocal Detection Approach for Named Entity Recognition and Mention Detection. ACL".

[5] Crichton, G., Pyysalo, S., Chiu, B. et al. "A neural network multi-task learning approach to biomedical named entity recognition". BMC Bioinformatics 18, 368 (2017)

[6] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In Proceedings of the 22nd international conference on World Wide Web, pages 13–24.

[7] ACM.Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. InProceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, Santa Fe,New Mexico, USA, August. Association for Computational Linguistics.