SkillSpotter: Mining and Skills from Job Descriptions using NER

Kabir Thakur
Syracuse University
kathakur@syr.edu

Pankaj Yadav
Syracuse University
pyadav05@syr.edu

Abstract

This paper presents "SkillSpotter", a novel approach to extracting and categorizing skills from job descriptions using advanced text mining techniques. Utilizing Named Entity Recognition (NER), the study leverages a fine-tuned BERT model that accurately identifies both general and specific technological and soft skills from a vast dataset of web-scraped job descriptions. The project addresses challenges in data sampling, annotation, and model optimization while considering the ethical implications of data usage and technology in human resource contexts. The findings offer valuable insights into job market trends and skill demands, highlighting the potential of NER in job recommendation systems.

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

1. Introduction

In this paper, we introduce 'SkillSpotter', a state-of-the-art text mining tool engineered to spot job skills from job descriptions. SkillSpotter harnesses advanced Natural Language Processing techniques, particularly Named Entity Recognition (NER) and a finely tuned BERT model, to analyze the evolving landscape of job requirements. To fine-tune the BERT model we created BIO-tagged training data of sentences from a dataset of web-scraped job descriptions.

Further augmenting our approach, we 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, our study zeroes in on 34 technology-based job titles, selected for their relevance to our research objectives. For each of these job titles, we meticulously maintain a distinct taxonomy, ensuring a tailored and nuanced understanding of the skill requirements.

In the following sections, we describe the methodology behind our approach, including data cleaning, sampling, and transformation followed by model training and inference. We then evaluate the performance of our proposed models and discuss their strengths, weaknesses, and potential improvements. Finally, we present our conclusions and discuss some use cases of our model. By investigating these advanced techniques, we aim to contribute to the ongoing development of more effective and personalized recommendation systems.

2. Related Work

In the past Duyet et al. [3] described a 'Skill2Vec' architecture which employs a unique embedding method to represent skills in a multi-dimensional space, facilitating the identification of relationships between different skills. It primarily utilizes a Word2Vec-inspired neural network approach, emphasizing vector space representation of skills. This approach is analogous to our use of BERT for context-aware skill extraction from job descriptions which leverages NER with a focus on entity identification.

In recent years, most developments in skill extraction, such as those presented in "SkillNER: Mining and Mapping Soft Skills from Any Text" by Silvia Fareri et al [2] have not employed transformer architectures, opting instead for methods

like Multi-Layer Perceptrons (MLP). While MLPs are effective, the use of BERT, as demonstrated in our approach, provides a contextual understanding of whole sentences rather than just individual words. This distinction is crucial as it allows for a more nuanced and accurate extraction of skills, particularly in complex job descriptions where context plays a significant role.

3. Method

The methodology of the SkillSpotter project involves a structured approach to mining skills from job descriptions using Named Entity Recognition as shown in Figure 1. This process begins with rigorous data cleaning to ensure the relevance and quality of the text. Subsequent steps include tokenization and BIO tagging, essential for preparing the text for NER. The core of the methodology is the application of the Distilbert-base-uncased model, specifically chosen for its effectiveness in NER tasks and lightweight. The model is meticulously trained and evaluated to ensure its accuracy in identifying and categorizing skills.

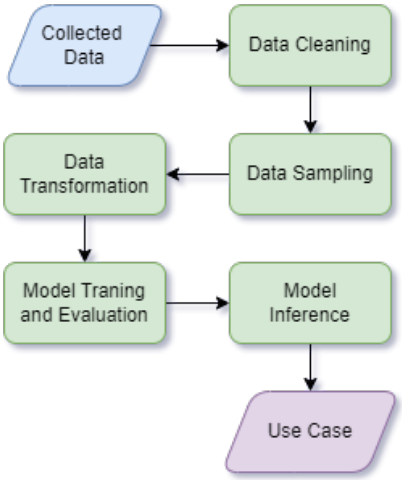

Figure 1. Methodology Flowchart.

3.1. Data Description and Exploratory Data Analysis

We used a dataset of 180K web-scrapped job descriptions from various websites. Each job description had a unique Job ID and a column for skills that were explicitly mentioned in the job description. The dataset included a lot of tech and non-tech job titles for jobs based in the United States.

In the dataset, we had about 87K unique job titles. Most of these titles were vague like 'Team Member' and 'Crew Member' and a lot of job titles were from the healthcare industry.

For creating the skills taxonomy we used the approach shown in Figure 3. We created a master list of skills by combining technical and soft skills from O*Net OnLine and added all the unique skills mentioned explicitly in the job descriptions.

3.2. Data Cleaning and Sampling

The data cleaning and sampling process was meticulously structured. We used the BeautifulSoup 4 library to parse the HTML. We 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. We also removed some noticeable incorrect annotations like empty quotes and words like 'auto'.

We 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 we had at least 30 job descriptions for each title.

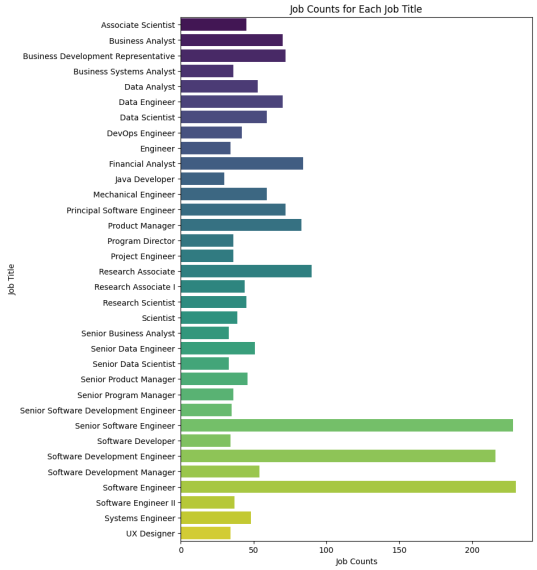

Figure 2. Job counts for each job title.

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. We started by

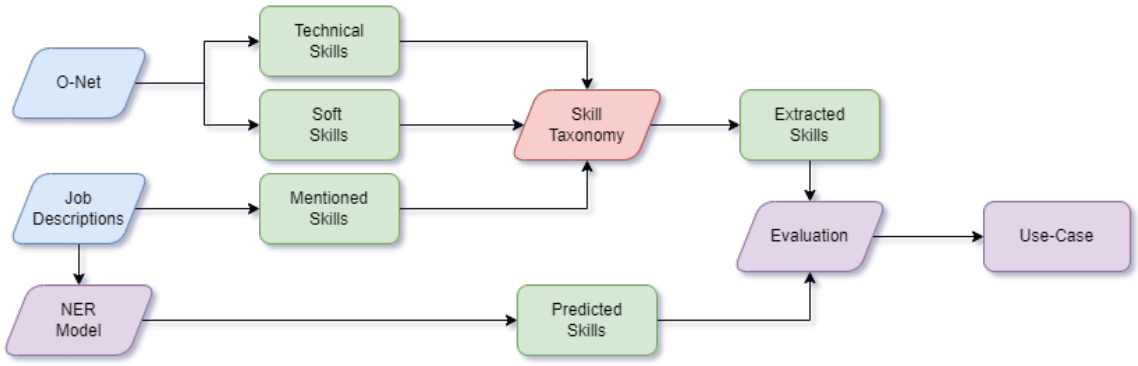

Figure 3. Skills Taxonomy Flow Diagram.

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, we flattened all job descriptions into a dataset of sentences and their corresponding tags and removed all sentences where there were no phrases present.

We used the distilbert-base-uncased tokenizer to again tokenize our 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 our model and also led to a misalignment in the BIO tags. Thus we 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

The final transformed dataset consisted of 18K sentences which were split into training and testing samples on an 80:20 ratio. We fine-tuned the distilbert-base-uncased transformer model with our training data. We chose Distilbert uncased as it is a lighter and faster version of the full BERT model. Distilbert uncased has been trained to mimic the probabilities of the full BERT model, with a small architecture that has 40% less parameters, and runs 60% faster while preserving over 95% of BERT's performance [1].

We trained for 3 epochs, with a constant learning rate of 0.00002 and a weight decay of 0.01. Each training epoch ran for 40 minutes. The model achieved an accuracy of 98.98% and a maximum F1 score of 93.78%. From Figure 6 we can see that the training and validation loss both decrease right after the start of training. The rate of decrease of training loss slows down after the second epoch while the validation loss completely converges.

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

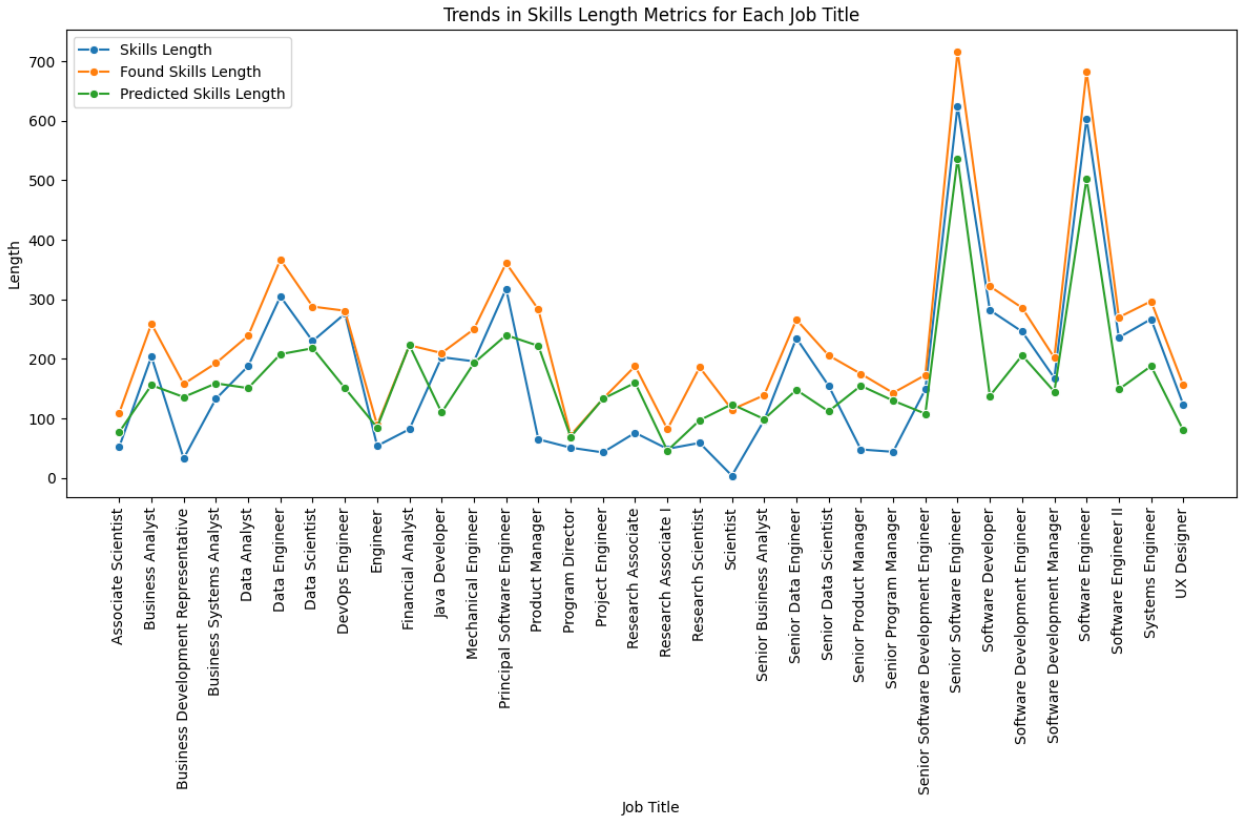

Figure 4. Trends in skill length metrics for each Job Title.

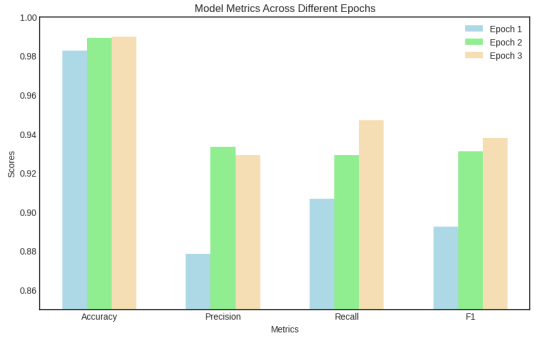

Figure 5. Model Metrics.

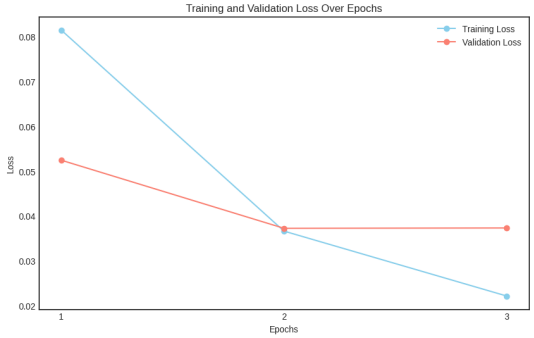

Figure 6. Training and Validation loss variation.

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

An F1 score of 93.78% is quite high, indicating a strong balance between precision (the model's ability to label as positive only those samples that are actually positive) and recall (the model's ability to find all the positive samples). This suggests the model is not only accurate but also reliable in its predictions.

4.1. Model Inference

We tested the model on our 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.

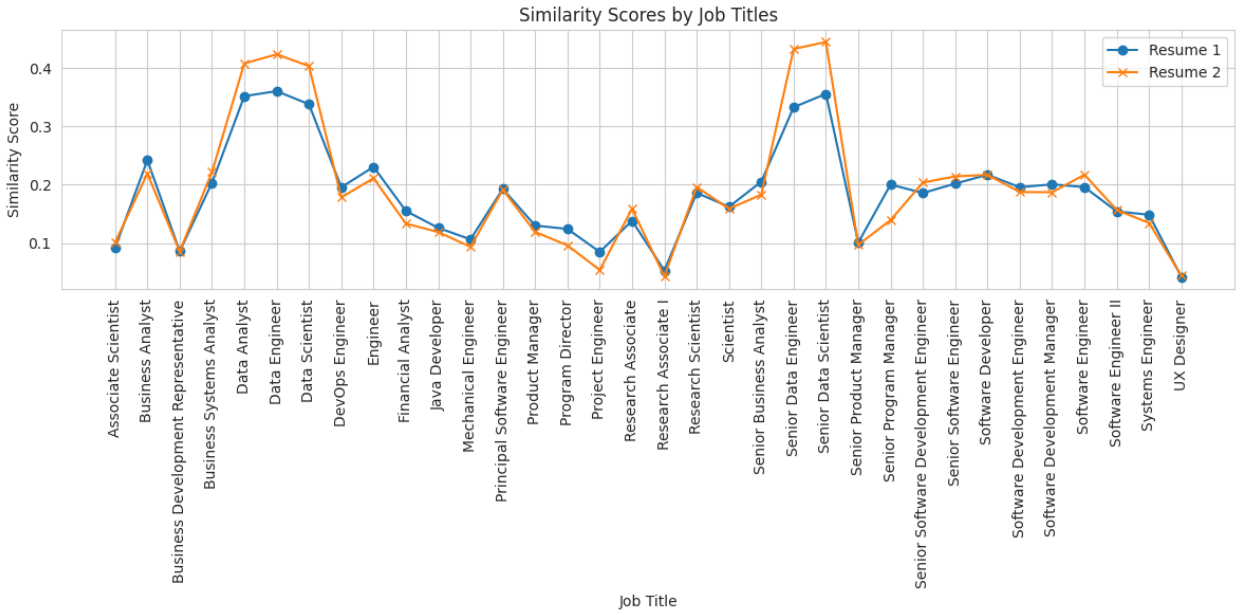

Figure 7. Similarity Score for 2 sample resumes for each Job Title.

We can see from Figure 4 that with pattern matching from our skill taxonomy, we were able to identify a lot more skills from the data than were mentioned explicitly in the job posting. Our model was able to match the results from pattern matching with almost 99% accuracy. It might not look like our model matched the results, but in Figure 4 we are aggregating for all the skills found per job title.

We tested our model to explicitly catch skill words from sentences generated by ChatGPT and it was able to identify known skill phrases and new soft skill phrases with high certainty but struggled a little with new skill words. The model was able to identify some abbreviations like JS for JavaScript but not ml for Machine Learning. We misspelled some skill words by a letter and the model was able to identify 7 out of 10 minor spelling mistakes.

Using the model we generated unique lists of skills for each job title to identify skills that were not present in other job titles. These skills have high feature importance for each job title class.

Using the data from Figure 2, we 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

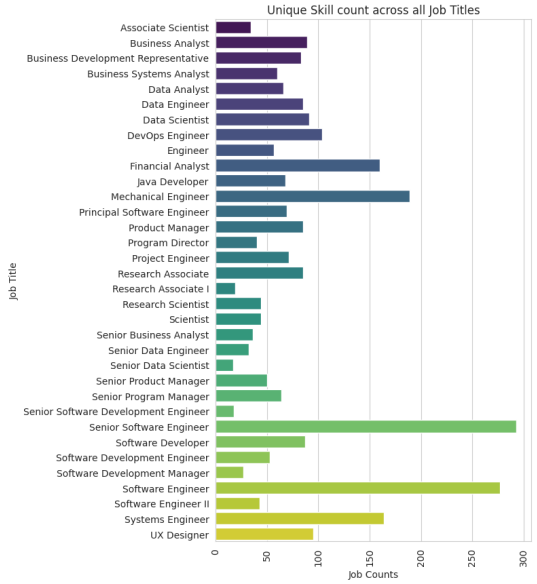

Figure 8. Unique skill count among all Job Titles

4.2. Use Case

This model can be used to extract skills from any text tokenized by the Distilbert tokenizer. In Figure 7

we tried using this model to extract skills from 2 sample resumes and calculated the cosine similarity between all skills required for each job title. This model can also be used as a recommendation system, recommending jobs to people with a high similarity score for skills present in the job description.

5. Conclusion

In conclusion, the model performed well, in about 15% of the cases our model was able to find new skills in job descriptions that were not present in our skill taxonomy. Thus we can say that the best technique for a recommendation system would be to maintain an adaptable skill taxonomy along with a BERT model to identify all the skills present in job descriptions and update the taxonomy whenever required.

6. Ethics Statements

Ethical Use of Open-Source Data

This data is open source and scraped from the web. We ensure 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

We have trained this model on a dataset which only contained Job Descriptions form companies in the US. We may be missing out on region specific skills if model is used to infer on a dataset with jobs from another geolocation. Similarly, we 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).