

1. Extract technical skills from a given resume text using spaCy PhraseMatcher.
2. Create a predefined skill list and match it against input text using spaCy.
3. Use spaCy NER to identify skill-related entities from job description text.
4. Normalize extracted skills by converting them to lowercase and removing duplicates.
5. Store extracted skills in a dictionary with `technical_skills` and `soft_skills` keys.
6. Merge skill outputs from spaCy Matcher and spaCy NER into a single list.
7. Implement logic to resolve conflicts such as abbreviations and full-form skills (e.g., ML and machine learning).
8. Use Sentence-BERT to calculate similarity between sentences and a master skill list.
9. Apply a similarity threshold to decide whether a sentence represents a valid skill.
10. Categorize merged skills into technical and soft skills using a mapping approach.
11. Build a function that accepts resume text and returns a unified skill list using spaCy and BERT.
12. Handle overlapping skills detected by multiple pipelines and keep only one standardized version.
13. Design a pipeline that prioritizes spaCy-based skills over BERT-based skills when both detect the same skill.
14. Generate a final structured output and save it as `final_skills.json`.
15. Compare the results of spaCy-only, BERT-only, and combined pipelines for skill extraction accuracy.