## 

1. 简历关键字提取
   1. PDF扫描
      1. 文本提取：使用适合的PDF解析工具（如PyMuPDF或pdfplumber）提取简历中的文本内容。
      2. 预处理：对提取的文本进行初步清洗，包括移除无意义的格式符号、修正乱码、删除多余的空行或空格等。
   2. 板块分析
      1. 教育板块
         1. 定位教育板块：通过搜索特定关键词（如“Education”）来识别教育板块的开始和结束。
         2. 提取学历&学科信息：使用spaCy根据预定义的学科与学历列表通过上下文分析提取学科名称
      2. 技能板块
         1. 定位技能板块：通过搜索关键词（如“Skills”或“Technical Skills”）来定位技能部分。
         2. 列表提取：假设技能以列表形式呈现，按逗号或分号等分隔符提取单个技能。
   3. 数据清洗与整理
      1. 统一格式：将所有提取的关键词转换为统一的大小写格式，以便后续处理。
      2. 去除噪音：删除关键词中可能存在的任何非字母数字字符。
      3. 同义词处理：将不同表达方式重复：对于每个板块提取的关键词的相同关键词统一为一个标准形式，例如将“BSc”和“Bachelor of Science”统一表示。
      4. 去列表，移除重复的条目，确保每个关键词只被计数一次。
      5. 把最后的学历学科信息和skill list放进json文件中
2. 工作简介关键词提取
   1. 通过CSM api获取job posting具体信息，将id，qualifications，prefer program放进JSON文件中，存储进GCS
   2. 通过部署在Hugging Face上的LLAMA2和特征工程获取skills列表以及prefer program列表
   3. 关键词列表清洗
   4. 将job id，prefer program list 和 skill list放进JSON文件中，储存进GCS
3. 职位图谱构建
   1. 读取JSON文件，通过Neo4j Aura创建带有权重的基于job id，prefer program 和 relevant skill的知识图谱
4. 匹配算法
   1. community detection
      1. 通过社区识别方法将图谱分成几个子图，从而限制搜索范围提升搜索效率
   2. node similarity
      1. 分析简历技能节点与知识图谱中技能节点的相似性，从而得到最相关的job id