**Challenges**

1. Different layout of resumes. Eg: some people would put the date in front of the title of the resume, some people do not put the duration of the work experience, or some people do not list down the company in the resumes. This makes the resume parser even harder to build, as there are no fix patterns to be captured.
2. Resumes can be spread across multiple pages.
3. Text corresponding to a heading can be any position. It can be at bottom left or right. So understanding layout is important.

**Stage 1 – Resume Parsing**

**Objective:**

**Extract the text corresponding to Experience and skills section**

1. **Directly from resume**

* Use resumes directly to get layouts out of it and extract required text.

eg: using apache tika for parsing ,pypdf2, pdfminer etc

* Define our own rules and regex and some basic models to get required text extracted.

**To parse pdf document**

* Pdfminer
* Apache tika
* Pdf2tree

**Useful links:**

* [*https://medium.com/analytics-vidhya/python-packages-for-pdf-data-extraction-d14ec30f0ad0*](https://medium.com/analytics-vidhya/python-packages-for-pdf-data-extraction-d14ec30f0ad0)

**Limitation:**

* It would not be able to handle resumes with different layout.

1. **Using OCR (Computer vision based approach)**

* Understanding the layout of document from images using some deep learning model. (eg: use layout parser)
* Selecting Region of interest (ROI) based on layout
* Extracting text from ROI (eg: use tesseract)
* Extracting required information from text

**Useful links:**

* [*https://towardsdatascience.com/analyzing-document-layout-with-layoutparser-ed24d85f1d44*](https://towardsdatascience.com/analyzing-document-layout-with-layoutparser-ed24d85f1d44)

**resume parser architecture**

pdf

images

Image pre processing

Layout analysis using deep learning model

(eg: Layout LM model)

Rule based engine

Extracted ROI text and summarize

**Approach 1: object detection + text extraction + classification**

**Steps:**

* Identify the region of interest bounding boxes from images. (using layout parser or other models) This boxes can represent two classes – headings, text.
* Convert each blocks to text using tesseract
* Match the heading block with similar text block using simple algorithms like naïve bayes or cosine similarity.

Example:

heading blocks:

Text

Description automatically generated

Logo

Description automatically generated

Text blocks:

Text

Description automatically generated

Text

Description automatically generated

**Approach 2: Using Layout LM model (Requires more research)**

General doubts?

* Experience can spread across multiple pages. Is it possible to handle this with layout LM?

For Layout LM model?

* need to explore how to annotate the images ?
* how to annotate the resumes?
* do we need to annotate each and individual word? Is it mandatory ?

**Text, letter

Description automatically generated**

Annotation

```

{'box': [67, 178, 104, 202], 'text': 'DATE:', 'label': 'question', 'words': [{'box': [67, 178, 104, 202], 'text': 'DATE:'}], 'linking': [[0, 2]], 'id': 0}

```

**Stage 2: Feature Generation and Classification**

**Our objective is to generate features mainly based on skills and experience.**

**Option1:**

* **NER (skills) on text data. NER on JD and compare both**

**Option2:**

* **Generate vector directly from text corresponding to experience and skills section. Generate vector from JD and compare both.**

**Complete Workflow**

Image processing

Images

pdf

Resume parser (Layout parser)

[model for extracting headings and text]

Headings & text

Tesseract

ROI

Key matching Store

Similarity

Job description

Feature vector from JD

Feature vector from resume

Naïve Bayes classifier for matching headings with text

NER model to extract suitable skills from text

NER model to extract suitable skills from JD