Instant Resume Evaluation Engine

Jinesh Dhruv
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
jad6566@g.rit.edu

***Abstract*—With the emergence of Job portal's like Linked-In, Indeed, Zip-Recruiter, Hire, etc. , job searching has become more convenient. The portals allow the job seeker to find all relevant jobs at one place. The job portals provide job seekers with easy-apply, apply via link and email to the recruiter type of facilities. These enable the job seeker to apply for many jobs in quick time which resulted in getting many applications for a single job posting. The companies shortlist candidates by parsing their resumes to match with job-specific criteria. The companies find it difficult to parse and extract the candidate's information due to the presence of different resume structures. Only those candidates get shortlisted whose resume is correctly parsed and which satisfy the job-specific criteria. Further, The job seekers fail to understand why their resume not getting shortlisted. Also, job seekers find it hard to identify whether the resume has all keywords (i.e. skills, experience, qualification, etc.) and does it meet the job description criteria which could be due to the content or format of the resume. This paper focuses on extracting candidate information from its resume which is in PDF format. The paper proposes a hybrid approach which uses content-based & layout-based techniques for resume parsing. The hybrid approach uses a blend of rule-based and segmentation-based techniques for effective resume parsing.**

***Index Terms*—Resume, Information Extraction, Segmentation, Rule-based**

I. Introduction

With the rapid growth and development in Internet usage, many people have started to use internet for shopping, selling, searching, news, banking, job searching etc. The presence of Job portal's like Linked-In, Indeed, Zip-Recruiter, Hire, etc. have made job searching easy. These portals have helped in connecting job seeker with many job postings. Gone are the days where the job seeker had to reach out to different companies for job inquiry. Now, many companies post job openings on their website and on the above job portals. The job portals now had many job postings from different companies. They invite job seekers to create a profile and provide some information about what type of job are they looking, job location preference, etc. The job portal then recommends the job seeker with job postings based on seeker portal profile. This way it connects the companies and job seekers. The job seeker then shortlists from the recommendation list and applies for the job with resume.

Resume consists of candidates education, publication, work experience, projects, achievements, skills, etc. type of information. The candidate makes use of different resume templates to store this information. Many candidates prefer having the resume in PDF format as it can be opened on any operating system, formatting does not get messed up when opened on any platform and provides a better appearance. However, Existing resume parsing system still has trouble parsing and extracting the content from the PDF due to the presence of graphics, images, icons and different content-structures.

If the companies parsing system is unable to parse the resume than the candidate will not get shortlisted for the interview call. Also, if the system is able to parse the resume but still the candidate might not get shortlisted due to the absence of keywords, experience, etc. on its resume required for this job. The job seeker does not know why it's resume is not getting shortlisted for the interview calls which could be due to the above reasons.

In this paper, we propose a hybrid approach which uses content-based and layout-based techniques for effective resume parsing. The layout-based techniques are used to identify segments and content-based techniques are used to identify and extract the raw data from the resume. The segments are the categories in which the resume content gets divided. The major categories are Education, Work Experience, Personal Information, Projects, Skills, and Achievements. After the information extraction, the system provides a resume feedback which enables the candidate to understand how it's resume is going to be interpreted by the existing resume systems. This will help the candidate to make changes if needed before applying for the job which will increase the chances of getting an interview call.

The paper is divided into several parts: a literature survey, system architecture, implementation, results and conclusions.

II. Related Work

Resume parsing deals with unstructured data where many efforts have been made in the past to make a robust system for information extraction. In past resume, extraction was focused mainly on Word, Text and HTML format. Duygu Celik [3] proposes information extraction from word document resume format using ontology-based techniques. They convert the DOCX format to HTML format, identified paragraphs using HTML tags, use the sentenceEnd algorithm to identify each sentence, generated word tokens to compare with resume ontology concepts. These ontology resume concepts replace words with ontology acronyms for better-extended meaning. Later, a rule-based technique is used to parse the entire resume.

They only use ontology resume concepts to directly extract features without generating segments. Their work is restricted to word documents only.

Hui Han et al. [2] implemented automatic metadata extraction from the documents by scanning the chunks of the documents line by line. This metadata extraction of research papers helps classification of documents which are used in digital libraries. This paper focuses on Support Vector Machine algorithm to perform automatic metadata extraction. In document metadata extraction, the text is labeled with corresponding meta tags. For feature extraction, word and line-specific features are used for representing the data. For the generation of word-specific features, word clustering methods are applied which groups the similar words and the cluster formed is used as a feature. This paper highlights two-step algorithm for classification of lines present in the document into a single class or multiple classes. The first step focuses on independent line classification where ten-fold cross-validation is performed to identify the feature vector that needs to be labeled as a particular class. The second step involves contextual SVM classification for the purpose of improvement of classification of each line. In this step, the class labels of N lines before and after the current line L are concatenated to a feature vector and this feature vector is then trained to identify the class label. Hui Han et al. evaluated their results by using various metrics such as precision, recall, F-measure, and accuracy. The line classification algorithm does not provide better results as it assumes that each feature is present in one single line which is not true for many resumes.

Chen [1] talks about considering layout and content based technique for PDF parsing. Techniques used for identifying block segments heavily rely on the PDF's layout information and not on the content. It makes use of Support Vector Machine model to classify the segments and Conditional Random fields for sequence tagging for feature extraction. The information they extract include Name, Email, Phone, University, Degree, Major, and Date. The results obtained from this technique were good for some features in Education segment but not that great in the Personal segment. Also, the paper does not provide clear information about how the data was labelled and evaluated. The resume is a semi-structured document, so rule-based system incorporated with above technique can give you promising results.

III. Methodology

We propose a hybrid system incorporated with layout-based and content-based techniques for resume parsing. As the figure 1 shows, The user uploads its resume for parsing. The uploaded resume gets stored in the MongoDB database. Now, we perform resume parsing for block segmentation and feature extraction. The extracted information from the resume is displayed to the user for validation of the extracted information. The resume parser block in the diagram is the heart of the system which ensures pre-processing, block segmentation and classification, feature identification and feature extraction.

TABLE I
Category Table

Segments	**Features**
Personal	Name, Address, Phone, E-mail, Profile Link
Education	University, Major, Degree, GPA, Year, Address, Coursework
Work Experience	Company Name, Position, Duration, Work Description
Academic / Personal Projects	Title, Description, Duration
Skills	Technical skills, Non-technical skills
Other	Awards, publications, thesis

The entire project can be found in this github repository https://github.com/jineshdhruv8/ResumeParser

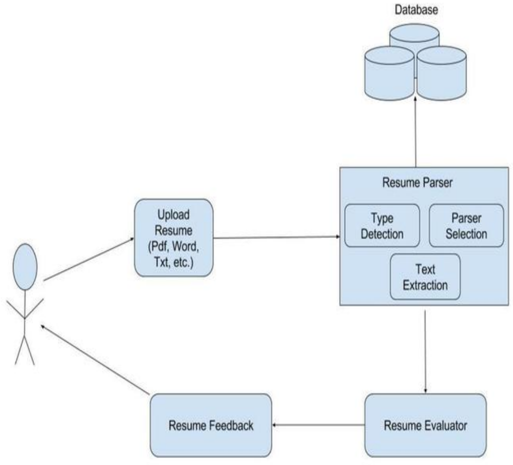

Fig. 1. Architecture

Initially, we divide the resume into block segments like personal segment, education segment, work experience segment, project segment, skill segment and another segment before feature extraction. This paper shows techniques only to extract features from personal and education segment. The Table 1 shows what features exist in each type of block segment.

Figure 2 shows the workflow of the resume parser block in detail. It consists of two main steps: 1) Block Segmentation and Identification and 2) Information Extraction. The raw PDF resume and parsed resume are both stored in MongoDB. In Block Segmentation we identify all the segments mentioned in figure 2. This is done using layout based and content-based information. The PDF layout information helps to identify segment location in the document and content that lies inside the segment. For information Extraction, we identify and extract features from personal and education segment. Figure 2 shows the list of features that are extracted from the above two segments.

A resume is a semi-structured document. To take advantage of this, we created wordlist that will help us in identifying this

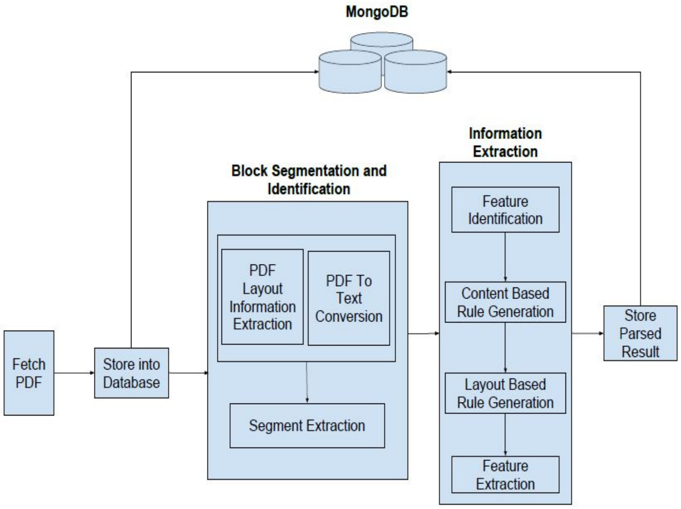

Fig. 2. Workflow

TABLE II
WORD LIST

Domain	Definition
Segment keyword list	List of all possible segment titles like education, qualification, university, project, personal project, academic project, etc.
University list	A university list in the US
US location list	A list of US Zip Codes, State and City
Degree list	A list of aliases & abbreviations of Masters, M.S., MS, Bachelor's, etc
Major list	A list of majors in United States
Company list	A list of all company in US

semi-structure for segmentation and feature extraction. Table 2 provides information about the wordlists used by the system for resume parsing. The next section will discuss in detail about the block segmentation and feature extraction steps.

A. Block Segmentation and Identification

In this step, we divide the resume into six segment category which is personal, education, work experience, project, skills and other segments. The personal segment consists of information regarding job seeker name, address, phone number, email, website and profile links. The education segment consists of information regarding university, major, degree, GPA, year, university address and coursework. The work experience segment consists of the company name, position, duration and work description. The project segment consists of project title, description, and duration. The skills segment consist of technical and non-technical skills. The other segment consists of awards, publication, and thesis.

The PDF document provides layout information like font size, font type, bounding box, and blank space alignment. The segment wordlist from figure 4 consist of keywords that help to identify segment boundaries. Each segment wordlist consists of all possible segment titles (e.g. For education segments, keywords are education, qualification, background, academic, university). We used the pdfminer package to read and parse PDF documents. We stored all the layout information for each sentence in the document and converted the PDF document to text format. Since the resume is a semi-structured document, we build some heuristic based rules for each segment (e.g. job seeker name is always at the top and has highest font-size in the document). These heuristic rules helped in determining segment boundaries. The fuzzy string matching was used for searching segment keywords of the wordlist in the document. Once we find the location of any segment keyword in the document then we find nearest neighbor (sentences) that are close to this location and store it in that segment. The neighboring sentences are validated before adding them to the respective segment to avoid miss-segmenting. The validation includes checking that the neighboring sentence should not include any other segment keywords and other heuristic rules. Below are steps involved for segmentation.

1) Store layout information(font size, bounding box, font style, blank space alignment) of the PDF document into dictionary
2) Convert PDF formatted resume to text format
3) Search segment list keywords into PDF document using fuzzy string matching algorithm
    a) If keyword present then store the position and fetch closest neighbor by distance formula
    b) If closest neighbor is empty or contains other segment keyword then find closest neighbor from the text format
4) Map each segment keyword in text document and start storing next sentences into the respective segment until the keywords of other segments are seen

B. Information Extraction

In this paper, we only present techniques to extract features from personal and education segment. The extracted feature of the personal segment is name, address, email, phone number and profile links. Figure 6 shows pseudo code for the feature extraction of the personal segment. To extract name feature we found sentences in a personal segment that are present at the topmost location and who have highest font size. Then we used natural language processing toolkit to identify proper nouns which gave us the name of the job seeker. If the name is not found then we check for words that are in uppercase or lowercase and are present at the top of the resume. We used different regular expressions to identify the phone number, email and profile links. To extract phone number, the algorithm search for numbers separated by brackets or dashes or spaces or no consecutive numbers of length greater than 4(e.g. (+1) 585-XXX-XXXX or +1585XXXXXXX or 585 XXX XXXX). To extract address we searched for zip-code of a 5-digit number and if found then use the wordlist dictionary to identify state and city. The profile links consist of the string starting with "https://" or "HTTP" or keywords like "LinkedIn" or "GitHub" and many more patterns. The regular expression used above patterns to search for profile links. Below are the steps to extract features from personal segment.

1) Find name:
   - For each sentence in personal segment:
     a) Find words that have highest font-size and position
   - Use NLP toolkit to identify proper nouns from the above shortlisted words
   - If name is still not found:
     a) Find words that are in uppercase or title-case and are present at the top of the resume
2) Find phone, email, link:
   - Scan all sentences near the position at which name feature was found and use regex to extract the number, email and profile links
3) Find address:
   - Scan all sentences near the position at which name feature was found and use regex to find the zip code
   - If Zip Code found then use wordlist to find city and state, else use NLP and wordlist to search for states and city

The extracted feature of education segment is university, major, date, degree, and GPA. Figure 7 shows pseudo code for the feature extraction of the personal segment. We created the wordlist of all university, all type of major and its abbreviations and all type of degree and its abbreviations to identify university, major and degree from the educations segment. To find university name, we search and match the university wordlist names with the education segment using fuzzy logic. If university name is not found then we use fuzzy logic to search for degree and major keywords in other segments and if found then search university name using above logic. The location where university name was found, we search degree, major, the date near that location. If university name is not found then we just search for degree and major keywords in the education segment using fuzzy logic.Below are the steps to extract features from education segment.

1) Find university:
   - For each sentence in Education segment until found:
     a) Find token of words that closely match with the university wordlist using fuzzy logic
   - If university not found:
     a) Search using above logic in other segments with some rules like degree or major keywords should also be found otherwise skip
2) Find degree, major and GPA:
   - Scan all sentences near the position at which university was found. Use fuzzy logic for string comparison to find degree abbreviations, words from the sentences using the wordlist
3) Find Date:
   - Scan all sentences near the position at which university feature was found and use regex and datefinder package to extract the date

IV. Results and Interpretation

A. Datasets

The dataset consisted of 50 PDF format resumes. These resumes were collected using crawler from Google Search Engine and randomly from the students of Rochester Institute Technology, North Eastern University, University of California, and Drexel University. The resume dataset consists of horizontal style and vertical style layout. In horizontal style layout, all the segments and its content are present below one another. In vertical style layout, the document is vertically divided into two or more sections and each section consist of some block segments and content aligned horizontally. For creating ground truth, we manually annotated block segments and features from each resume.

B. Evaluation Measures

To evaluate our system, we used precision, recall, and f-measure. Precision is the ratio of the number of relevant records retrieved to the total number of relevant and irrelevant records retrieved. For block segmentation, precision is the percentage of correct segments detected and retrieved to the total number of correct and wrong segments that were detected and retrieved. For feature extraction, precision is the percentage of correctly extracted feature that matches to the ground truth data. The recall is the ratio of the number of relevant records retrieved to the total number of relevant records present in the database. For block segmentation, recall is the percentage of correct segments detected and retrieved to the total number of correct segments present in the database. For feature extraction, recall is the percentage of correctly extracted feature to total features present in the ground truth data. F-measure is the weighted harmonic mean of the precision and recall.

C. Results

The block segmentation results can be seen in table 3 and feature extraction results can be seen in table 4.

TABLE III
Segmentation Results

Segment Detection	Precision	Recall	F1-score
Personal	97%	97%	97%
Education	76%	72%	74%
Work Experience	90%	75%	81%
Project	53%	47%	50%
Skills	75%	70%	72%
Other	41%	40%	40%

The personal segment does not require many keywords for identification compared to the education segment since it is always present at the top of the resume. Each segment has unique layout and content property which helps in identifying this segment. Our system identify all this segment-specific property and build content-layout based rules for segmentation

TABLE IV
FEATURE EXTRACTION RESULTS

Attributes	Precision	Recall	F1-score
Name	92%	90%	91%
Email	92%	90%	91%
Phone	92%	90%	91%
Profile Link	81%	77%	78%
Address	65%	57%	50%
University	88%	75%	80%
Major	70%	65%	77%
Date	55%	35%	42%
Degree	65%	60%	62%
GPA	60%	40%	48%

which produces better results. From the above two tables we can see that the accuracy of determining personal segment is better than the education segment.

The system produces poor result for project segmentation and other segmentation. One of the reason behind this result is the lack of keywords and weak layout rules. Both this segment has wide range of content and different layout styles which makes it difficult for the system for identification.

The features like name, email and phone are extracted with better accuracy compared to features like date, degree and GPA. The education segment consist of wide range of different major, date and GPA format. For example, major could be written as Masters in Computer Science or M.S. in Computer Science, MS in CS, etc. Similarly date could be present like Jun 17' - Sept 17' or 06/17-09/17 or June 2017 to September 2017, etc. The above are just few different feature representation of degree, major and date. To solve this issue, our system does pre-processing like stopwords and punctuation removal but still it does not identify each feature correctly. This is the reason why we get low accuracy for this features.

V. FUTURE WORK

To generate better segmentation result we plan to use and test different machine learning algorithm like Support Vector Machine and Naive Bayes classifier. For feature extraction we plan to research on Control Random Field algorithm for sequence tagging.

VI. CONCLUSION

In this paper, we propose a hybrid model which uses content-based and layout-based technique for parsing resume documents in PDF format. For resume parsing, the document is initially divided into six different segments. After segmentation, we do feature extraction using content-based and layout-based technique for feature extraction. This technique produces good segmentation and feature extraction results. The experimental result shows that the average segmentation F1-score of the model is 69% and the average feature extraction F-1 score is 71%.

REFERENCES

[1] J. Chen, L. Gao, and Z. Tang. Information extraction from resume documents in pdf format. In *Document Recognition and Retrieval*, 2016.

[2] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, pages 37–48, May 2003.

[3] D. elik, A. Karakas, G. Bal, C. Gltunca, A. Eli, B. Buluz, and M. C. Alevli. Towards an information extraction system based on ontology to match resumes and jobs. In *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, pages 333–338, July 2013.