Layout Aware Resume Parsing Using NLP and Rule-based Techniques

8th International Conference on Information Technology Research 2023

Layout Aware Resume Parsing Using NLP and Rule-based Techniques

- Mr. S.P. Warusawithana, Undergraduate, Faculty of Information Technology, University of Moratuwa
- Mr. N.N. Perera, Undergraduate, Faculty of Information Technology, University of Moratuwa
- Mr. R.L. Weerasinghe, Undergraduate, Faculty of Information Technology, University of Moratuwa
- Ms. T.M. Hindakaraldeniya, Undergraduate, Faculty of Information Technology, University of Moratuwa
- Dr. (Ms.) G.U. Ganegoda, Senior Lecturer, Faculty of Information Technology, University of Moratuwa

Overview

- Introduction
- Others Work
- Data Source
- Approach
- Implementation
- Evaluation
- Conclusions

Introduction

• Undergraduates in different fields are searching for chances in their own field and there is a competition between the candidates due to recent changes in the educational system.

• Undergraduates must go through various processes to construct a solid resume, even if it is a potent opportunity for a candidate. Reviewing the resume is the last and most difficult phase, requiring professional knowledge.

• However, since resumes are highly structured documents, it is not enough just to extract the content but need to be aware about the layout as well.

• Therefore, there is a need for a method which can help the candidate to get feedback for their resume by extracting section wise content using layout aware text extraction method using NLP and rule-based techniques

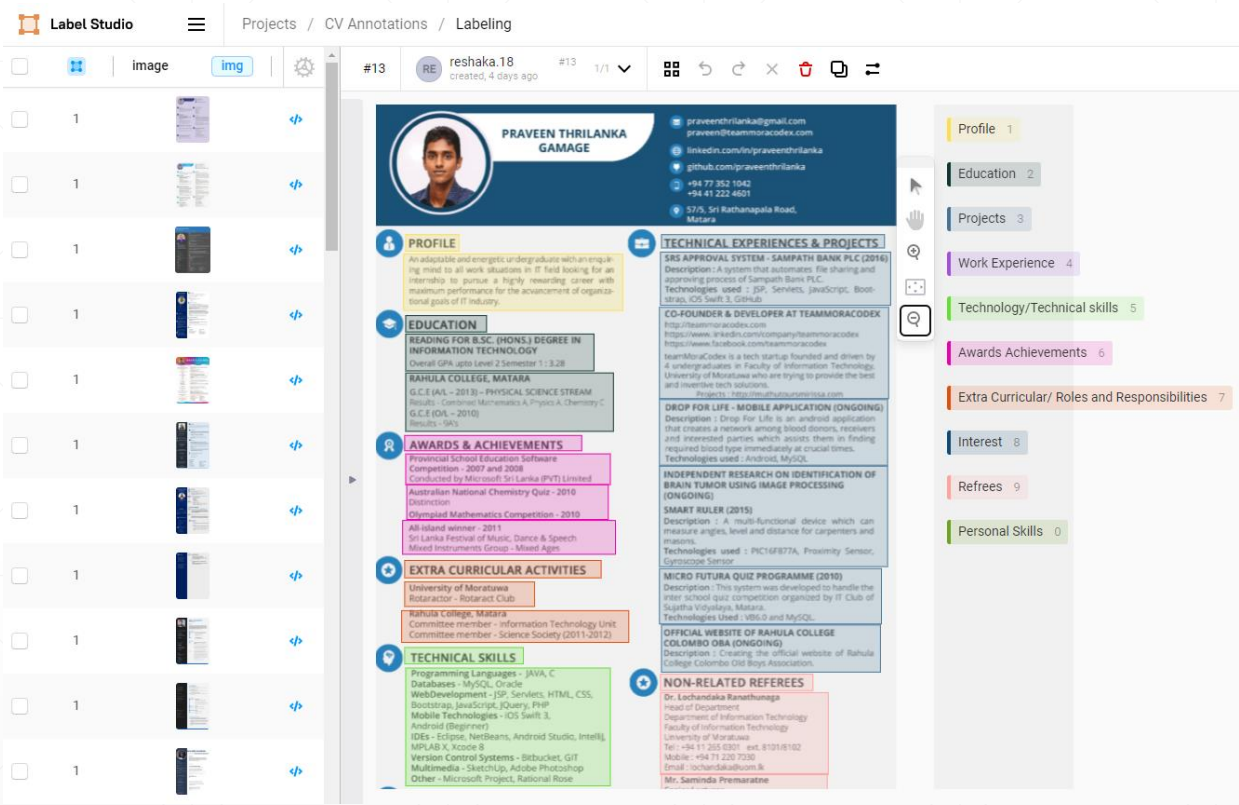
Others Work

- In the existing methods the system concentrates on the issue of data extraction from resumes in PDF format and suggests a hierarchical extraction methods, but the accuracy was found to be low, and the final output is not much valuable.
- Existing techniques have frequently ignored the crucial element of resume formatting in favor of extracting primary entities from resumes

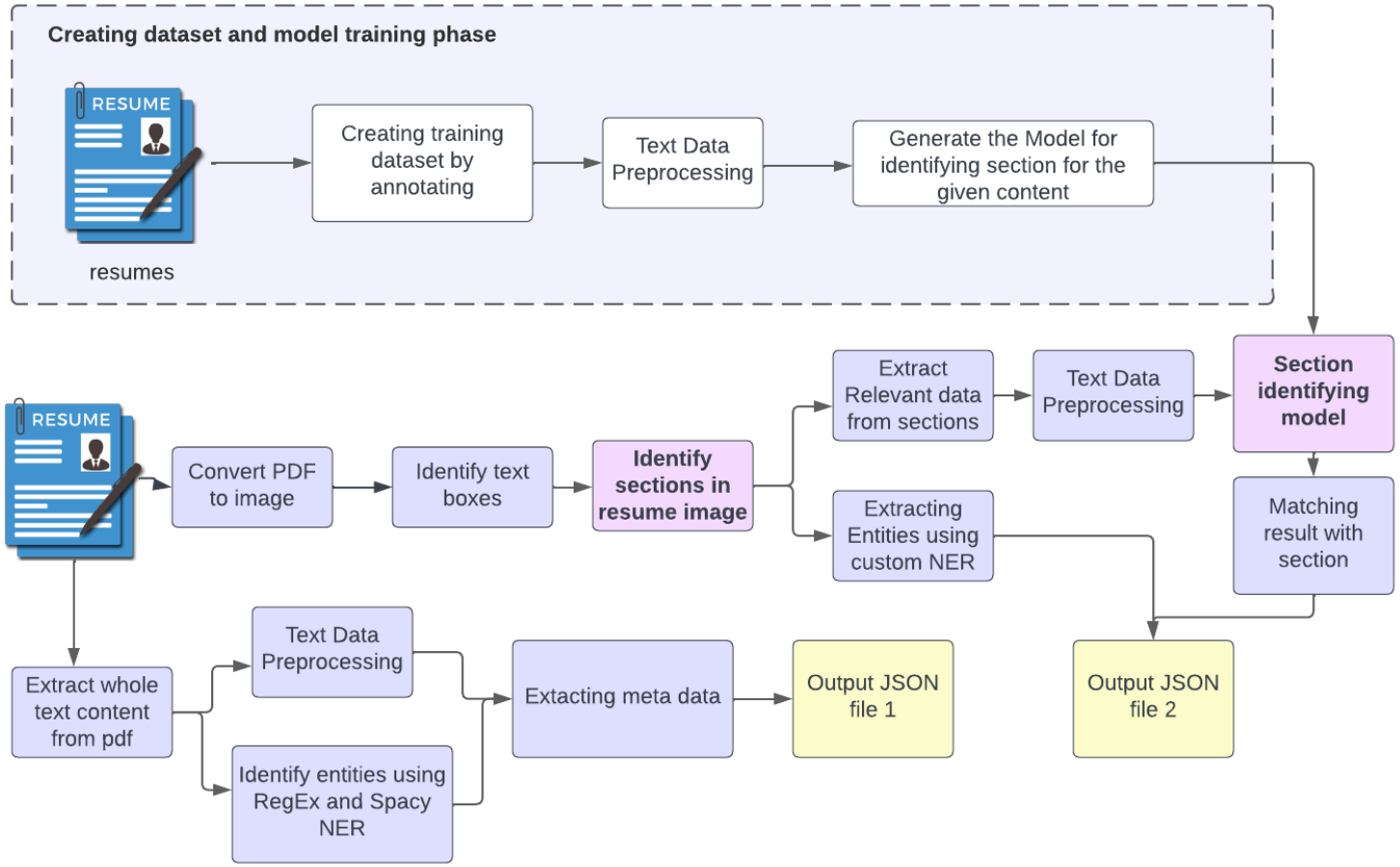
Dataset

- **The resume list and their details** including the selected company were obtained from the system administrator of the **industrial training platform** of Faculty of Information Technology, University of Moratuwa.

- **Manually annotated resumes** using label studio according to module requirements



Approach


Creating dataset and model training phase
RESUME
resumes
Creating training dataset by annotating
Text Data Preprocessing
Generate the Model for identifying section for the given content
RESUME
Convert PDF to image
Identify text boxes
Identify sections in resume image
Extract Relevant data from sections
Text Data Preprocessing
Section identifying model
Extracting Entities using custom NER
Matching result with section
Extract whole text content from pdf
Text Data Preprocessing
Identify entities using RegEx and Spacy NER
Extacting meta data
Output JSON file 1
Output JSON file 2

Implementation

Task 01 - Extracting all text content, entities using spacy NER, RegEx and meta data from resume for other modules.

This part is mainly focus to identify important entities such emails, social accounts links, phone numbers and other urls.

```
pdf_doc = nlp(extract_text_from_pdf('11.pdf'))

final_text = str(pdf_doc)

def clean_text(text):
    text = text.lower()
    text = text.replace("\"","")
    text = text.replace("\'","")
    string_encode = text.encode("ascii", "ignore")
    string_decode = string_encode.decode()
    return string_decode

final = clean_text(final_text)

content  = " ".join(final.split())
print(content)

# print("**********")
# getting emails using spacy
emails = []
for token in pdf_doc:
    if token.like_email:
        emails.append(str(token))
print(emails)

# # getting links using spacy
otherUrls = []
for token in pdf_doc:
    if token.like_url:
        otherUrls.append(str(token))
print(otherUrls)

#getting phone numbers
phone_numbers = re.findall(r'(?:\+\d{1,3}\s?)?(?:(?:\(?:\d{3}\)\s?)|(?:\d{3})(?:\s|-?))(?:\d{3}(?:\s|-?))(?:\d{4})(?:\s?(?:(?:[E
print(phone_numbers)

# print("**********")
```

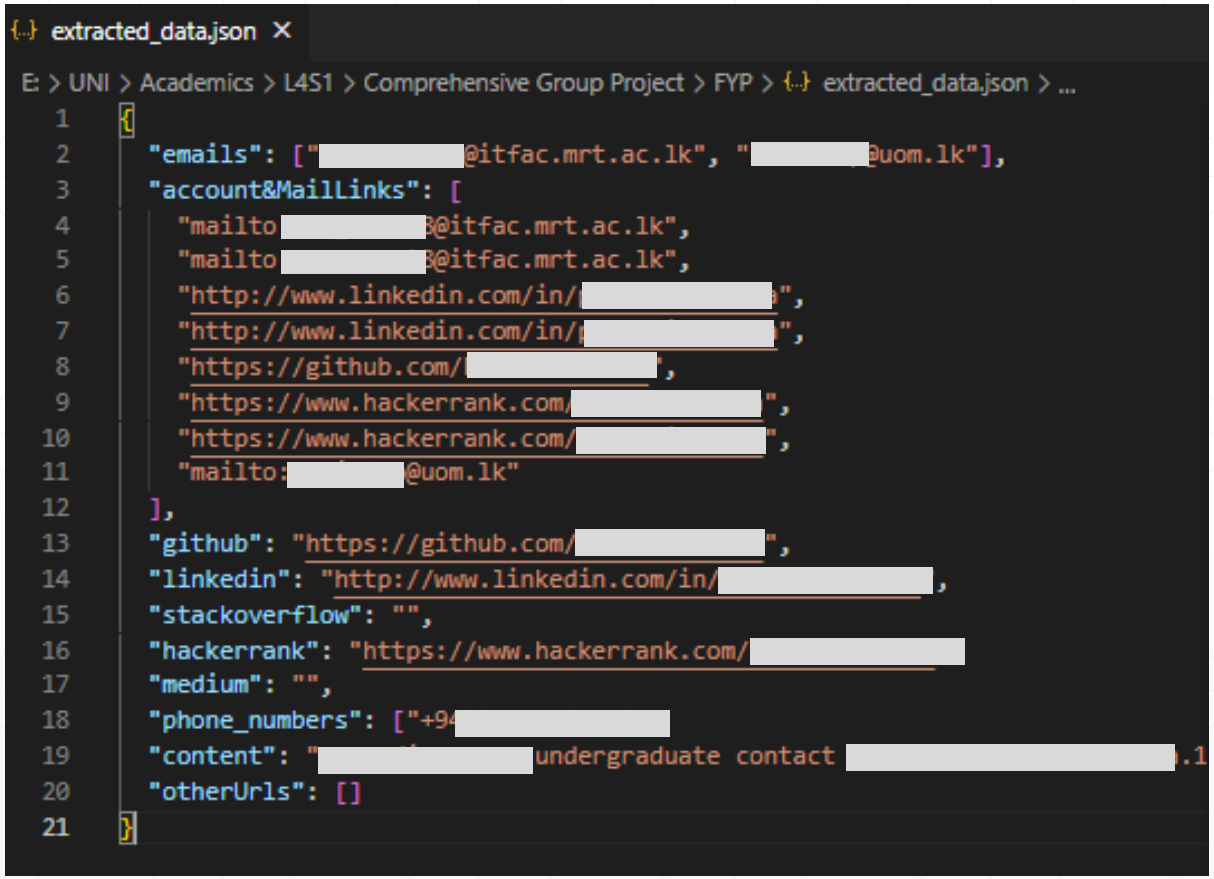
Extract emails, phone numbers and other urls

```
#getting accounts links from meta data

import fitz

linkedin = ''
github = ''
stackoverflow = ''
hackerrank = ''
medium = ''

urls = []

with fitz.open('11.pdf') as my_pdf:
    for page_number in range(1, len(my_pdf)+1):
        page = my_pdf[page_number-1]

        for link in page.links():
            if 'uri' in link:
                print(link['uri'])
                urls.append(link['uri'])
                if re.findall(r'http[s]?:\/\/www\.linkedin\.com\S+|www\.linkedin\.com\S+|http[s]?:\/\/linkedin\.com\S+|linkedin.
                    linkedin = link['uri']
                elif re.findall(r'http[s]?:\/\/www\.github\.com\S+|www\.github\.com\S+|http[s]?:\/\/github\.com\S+|github.com\S+
                    github = link['uri']
                elif re.findall(r'http[s]?:\/\/www\.stackoverflow\.com\S+|www\.stackoverflow\.com\S+|http[s]?:\/\/stackoverflow\.
                    stackoverflow = link['uri']
                elif re.findall(r'http[s]?:\/\/www\.hackerrank\.com\S+|www\.hackerrank\.com\S+|http[s]?:\/\/hackerrank\.com\S+|ha
                    hackerrank = link['uri']
                elif re.findall(r'http[s]?:\/\/www\.medium\.com\S+|www\.medium\.com\S+|http[s]?:\/\/medium\.com\S+|medium.com\S+
                    medium = link['uri']
```

Get meta data and find account links using regex

Implementation

Output of the first task



Json file with identified entities

Implementation

Task 02 – Identify different paragraphs or sections from the resume.

Main objective of this task is to distinctly identify paragraphs or sections from the resume when user upload their resume.

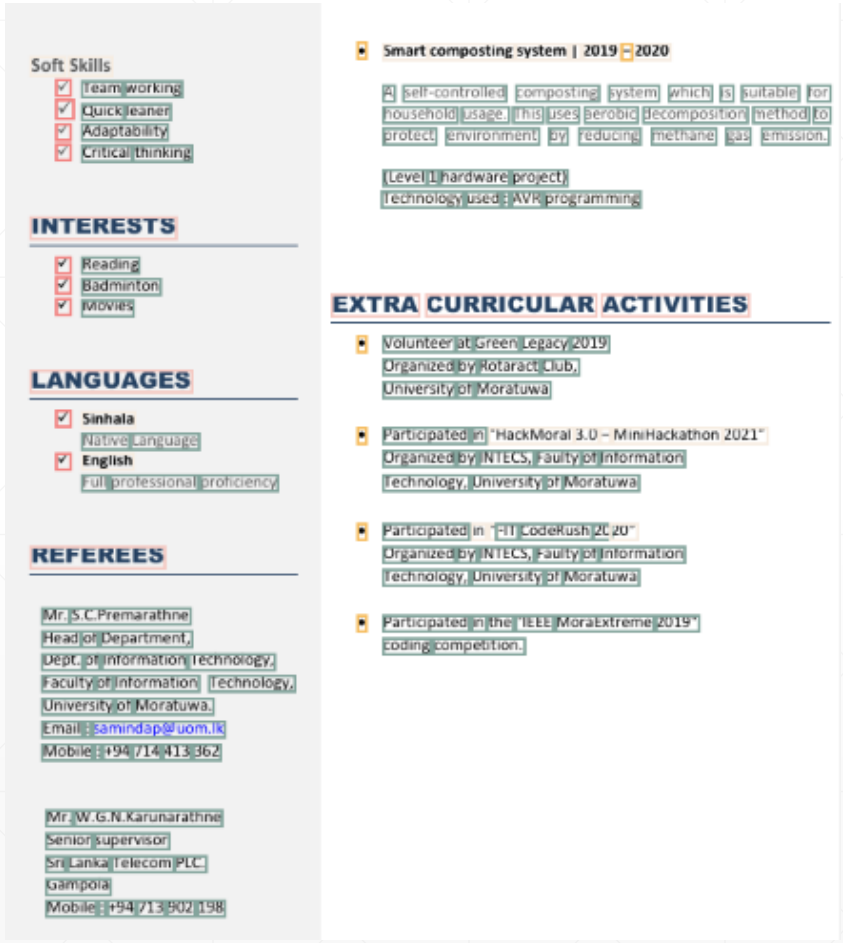
```
pdf_tokens, pdf_images = lp.load_pdf(pdf_path, load_images=True)

text = ''
lines = []

for i in range(len(pdf_tokens[0])):
    block = pdf_tokens[0][i]
    next_block = None
    if not i+1==len(pdf_tokens[0]):
        next_block = pdf_tokens[0][i+1]

    if text == '':
        text = block.text

    if not next_block == None and (abs(next_block.block.x_1 - block.block.x_2) < 10 or abs(next_block.block.y_1 - block.block.y_
        text += (' ' + next_block.text)
    else:
        text = ''

lp.draw_box(pdf_images[0], pdf_tokens[0])
```

Drawing bounding boxed for words.

```
block_sets = []

first_block = None
last_block = None
x_first = 0
x_last = 0
y_last = 0

for y in range(len(pdf_images)):
    temp_block_sets = []
    for i in range(len(pdf_tokens[y])):
        block = pdf_tokens[y][i]
        next_block = None

        if not i+1==len(pdf_tokens[y]):
            next_block = pdf_tokens[y][i+1]

        if text == '':
            text = block.text
            first_block = block
            x_first = block.block.x_1
            x_last = block.block.x_1
            y_last = block.block.y_1

        if not next_block == None and (abs(next_block.block.x_1 - block.block.x_2) < 10 or abs(next_block.block.y_1 - block.bloc
            text += (' ' + next_block.text)
            if next_block.block.x_1 < x_first:
                x_first = next_block.block.x_1
            if next_block.block.x_2 > x_last:
                x_last = next_block.block.x_2
            if next_block.block.y_2 > y_last:
                y_last = next_block.block.y_2
        else:
            first_block.block.x_1 = x_first
            first_block.block.x_2 = x_last
            first_block.block.y_2 = y_last
            first_block.text = text
            first_block.page = y
            block_sets.append(first_block)
            temp_block_sets.append(first_block)
            text = ''
```

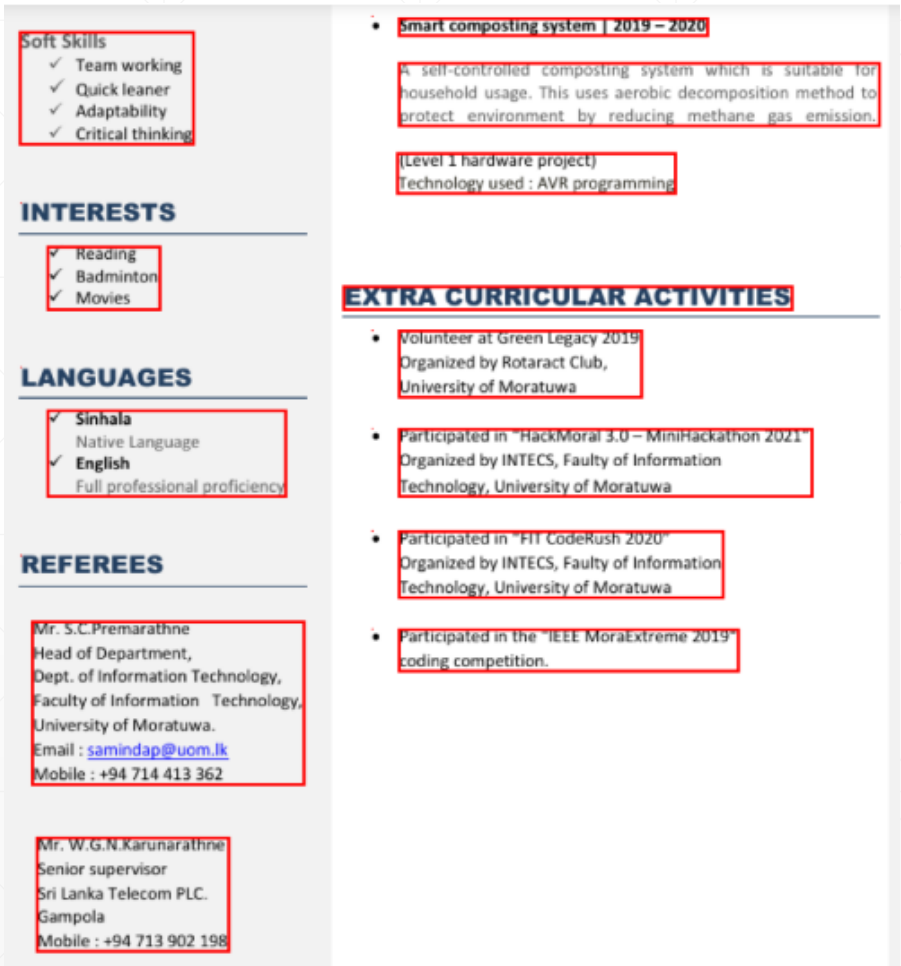
Identifying distinct sections using spaces

Implementation



Drawing Bounding Boxes

Applying the approach



Identify distinct paragraphs

Implementation

Task 03 – Implementing a multiclass classification model to predict the resume section.

Main objective of this task is to identify the relevant resume section after identifying the distinct paragraphs from task 02.

- Profile
- Education
- Projects
- Technical Skills
- Personal Skills
- Awards, Achievements and Responsibilities
- Interests
- Referees

Implementation

Task 03.01 - Generate data Set for multiclass classification model

Label studio is the software which is used to annotate the data. It gives a coco file of the annotated data and from that coco file we can generate the csv.


```
"annotations": [
  {
    "id": 0,
    "image_id": 0,
    "category_id": 5,
    "segmentation": [],
    "bbox": [
      123.30170289913511,
      438.62370646675316,
      164.94703849205027,
      33.74276442733343
    ],
    "ignore": 0,
    "iscrowd": 0,
    "area": 5565.769062823551
  },
```


Position coordinates of the annotations

```
ocr_agent = lp.TesseractAgent(languages='eng')
data = []
count = 0

main_folder = 'All Annotated CVs'
folders = ['Folder 1-3, 18', 'Folder 4-8', 'Folder 9-13', 'Folder 13-17']

for folder in folders:
    COCO_ANNO_PATH = main_folder + '/' + folder + '/result.json'
    COCO_IMG_PATH  = main_folder + '/' + folder + '/'

    coco = COCO(COCO_ANNO_PATH)

    for image_id in coco.imgs.keys():
        image_info = coco.imgs[image_id]
        annotations = coco.loadAnns(coco.getAnnIds([image_id]))
        path = COCO_IMG_PATH+'images/'+image_info["file_name"][9:]
        image = cv2.imread(f'{path}')
        layout = load_coco_annotations(annotations, coco)
        for block in layout:
            segment = block.crop_image(image)
            block.text = ocr_agent.detect(segment)

        image_id = image_info["id"]
        count = count + 1
        print(folder, ' - ', count)

        file_id_array = image_info["file_name"].split('-')
        array_seg1 = file_id_array[1].split('_')
        image_folder = array_seg1[0][2:]
        cv_number = array_seg1[1]
        page = file_id_array[2][3:4]

        for text in layout:
            related_text = text.text.replace('\n',' ')
            data_item = [image_folder, cv_number, page, related_text, text.type]
            data.append(data_item)

all_data = pd.DataFrame(data, columns=['Folder','Id','Page','Text', 'Section'])
all_data.to_csv(main_folder + '/Annotated_data.csv')
```

Creating the csv file from coco file

Implementation

Annotating data manually

Text	Section
Dedicated third year undergraduate with strong interpersonal skills a	Profile
ABOUT ME	Profile
Dedicated third year undergraduate with strong interpersonal skills a	Education
EDUCATION	Education
B.SC. (HONS.) IN INFORMATION TECHNOLOGY UNIVERSITY OF MORAT	Education
HOLY FAMILY CONVENT, COLOMBO 04 G.CE (A/L - 2013) - PHYSICAL SC	Education
PROFESSIONAL QUALIFICATIONS	Education
Successfully completed a Certificate course in Computer Science at N	Education
AWARDS & ACHIEVEMENTS	Awards Achievements
Dean's list in Level 1 Semester 2	Awards Achievements
EXTRA CURRICULAR ACTIVITIES	Extra Curricular/ Roles and Responsibilities
University of Moratuwa Rotaractor - Rotaract Club (2016-Present)	Extra Curricular/ Roles and Responsibilities
Holy Family Convent Vice President- Buddhist Union (2011) Member	Extra Curricular/ Roles and Responsibilities
NON-TECHNICAL PROJECTS	Extra Curricular/ Roles and Responsibilities
The Exemplar- Are You Ready? 2016 Session co-chair person Organize	Extra Curricular/ Roles and Responsibilities
TECHNICAL SKILLS	Technology/Technical skills
Programming Languages - C, Java Databases - MySQL, Oracle Web de	Technology/Technical skills

Generated Dataset

Implementation

Task 03.02 – Implementing the multiclass classification model using dataset

The main objective of this sub task is to implement the multiclass classification model to predict the relevant section for given content.

Before using a classification algorithms, the system clean the text content by removing unnecessary parts. And then pipeline was used to train the model which includes the CountVectorizer, TfidfTransformer and the Support Vector Machine with SGD Classifier.

- Regression Algorithms
- Multinomial Naive Bayes
- Gaussian Naive Bayes
- Support Vector Machines

```
from sklearn.model_selection import train_test_split
X = df.Text
y = df.Section
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 42)
```

```
#svm model
from sklearn.linear_model import SGDClassifier

text_clf_svm = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer()),
                     ('clf-svm', SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3, random_state=42))
                     ])

text_clf_svm = text_clf_svm.fit(X_train, y_train)

y_pred_svm = text_clf_svm.predict(X_test)

print(f'accuracy {accuracy_score(y_pred_svm,y_test)}')
```

```
accuracy 0.8881431767337807
```

Building the model

```
test_data ="An enthusiastic and passionate individual who like to work with
cleaned_text = clean_text(test_data)

print('Predicted Section - '+ text_clf_svm.predict([cleaned_text])[0])
```

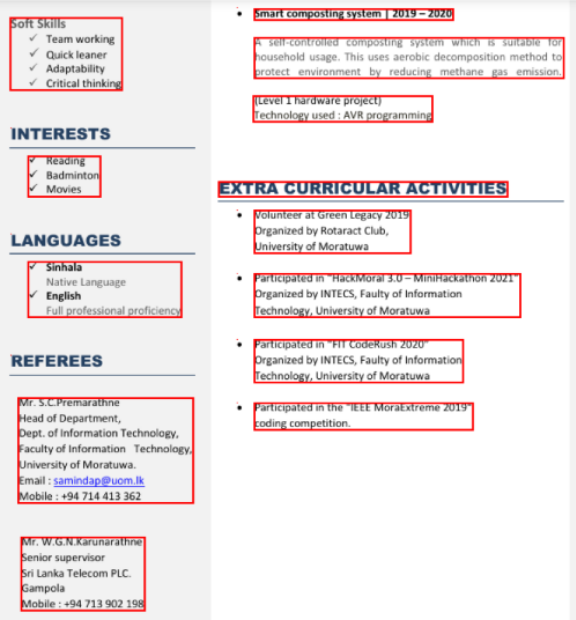
```
Predicted Section - Profile
```

How the model predict relevant section

Implementation

Task 04 – Extracting the section wise data using model and custom NER.

Main objective of this task is to extract all the section wise text content using the multiclass classification model which is implemented in previous task and in addition to that use a custom NER to identify important entities.



Soft Skills
- Team working
- Quick leaner
- Adaptability
- Critical thinking

INTERESTS
- Reading
- Badminton
- Movies

LANGUAGES
- Sinhala
  Native Language
- English
  Full professional proficiency

REFEREES

Mr. S.C.Premarathne
Head of Department,
Dept. of Information Technology,
Faculty of Information Technology,
University of Moratuwa.
Email : samindap@uom.lk
Mobile : +94 714 413 362

Mr. W.G.N.Karunarathne
Senior supervisor
Sri Lanka Telecom PLC.
Gampola
Mobile : +94 713 902 198

- Smart composting system | 2019 – 2020

A self-controlled composting system which is suitable for household usage. This uses aerobic decomposition method to protect environment by reducing methane gas emission.

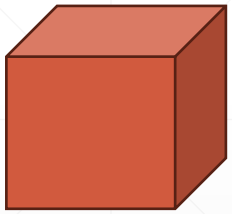
(Level 1 hardware project)
Technology used : AVR programming

EXTRA CURRICULAR ACTIVITIES

- Volunteer at Green Legacy 2019
  Organized by Rotaract Club,
  University of Moratuwa
- Participated in "HackMoral 3.0 – MiniHackathon 2021"
  Organized by INTECS, Faulty of Information Technology, University of Moratuwa
- Participated in "FIT CodeRush 2020"
  Organized by INTECS, Faulty of Information Technology, University of Moratuwa
- Participated in the "IEEE MoraExtreme 2019" coding competition.

Sending each ane every paragraph content through the multiclass classification model

Classification model



Output the predicted section

Implementation

After sending all the distinct paragraph through the implemented model, then the system divide those content to sections. Additionally, the system used a pre-trained custom Named Entity Recognition (NER) model to extract significant entities from the resume.

```
for txt in block_sets:
    cleaned_text = clean_text(txt.text)
    pre = loaded_model.predict([cleaned_text])[0]

    if pre == 'Profile':
        if ((cleaned_text=="profile") or (cleaned_text=="about me") or (cleaned_text=="objective") or (cleaned_text=="summary")):
            profileContent.append(cleaned_text)
            profileContentBox.append(txt)
        elif (len(cleaned_text.split(" ")) > 5):
            profileContent.append(cleaned_text)
            profileContentBox.append(txt)
    elif pre == 'Education':
        if ((cleaned_text=="education") or ("gpa" in cleaned_text) or ("college" in cleaned_text)):
            educationContent.append(cleaned_text)
            educationContentBox.append(txt)
        elif (len(cleaned_text.split(" ")) > 5):
            educationContent.append(cleaned_text)
            educationContentBox.append(txt)
    elif pre == 'Technology/Technical skills':
        if ((cleaned_text=="language") or (cleaned_text=="personal")):
            continue
        technicalSkillsContent.append(cleaned_text)
        technicalSkillsContentBox.append(txt)
    elif pre == 'Personal Skills':
        personalSkillsContent.append(cleaned_text)
        personalSkillsContentBox.append(txt)
    elif pre == 'Interest':
        interestContent.append(cleaned_text)
        interestContentBox.append(txt)
    elif (pre == 'Extra Curricular/ Roles and Responsibilities') or (pre == 'Awards Achievements'):
        if (len(cleaned_text.split(" ")) > 3):
            awardsResponsibilitiesContent.append(cleaned_text)
    elif pre == 'Refrees':
        if (("referees" in cleaned_text) or ("referee" in cleaned_text)):
            tempRefereesContent.append(cleaned_text)
            tempRefereesContentBox.append(txt)
        elif (len(cleaned_text.split(" ")) > 4):
            tempRefereesContent.append(cleaned_text)
            tempRefereesContentBox.append(txt)
```

Getting the predicted section for every distinct paragraph

Sachitha Ilayperuma PERSON NAME UNDERGRADUATE Faculty of Information Technology ORGANIZATION University of Moratuwa EDUCATION 2017 2016 University of Moratuwa, Sri Lanka: Faculty of Information Technology Reading for Bachelor of Science (Hons.) in Information Technology DEGREE (2017-2021) GPA: 3.1135 GPA (Up to third semester) Richmond Collage Galle G.C.E. Advanced Level Common Stream (2016) Results: Co-Mathematics A, ICT A, Physics B, Z-score 2.1684 PROJECTS COMPLETED Uluwitike, Galle, Sri lanka. 2019 (+94) 778738593 CONTACT NO snilayperuma@gmail.com EMAIL https://www.linkedin.com/in/ sachitha-ilayperuma/ ABOUT ME I am a hardworking PROFILE DESCRIBING ADJ and ambitious PROFILE DESCRIBING ADJ individual with a great passion for the IT industry. I am quick to adapt to work and capable of working in a team environment. I have good communication skills which enables effective communication. I am seeing an intern DESIGNATION position in the industry to practice and enhance my knowledge, skills and experience, ultimately contributing to the operations of the organization that I work for. PROFESSIONAL SKILLS Java PROGRAMMING LANGUAGES (TECH SKILLS) C PROGRAMMING LANGUAGES (TECH SKILLS) PHP WEB DEVELOPMENT (TECH SKILLS) MySQL DATABASE (TECH SKILLS) MSSQL DATABASE (TECH SKILLS) Angular WEB DEVELOPMENT (TECH SKILLS) Spring Boot WEB DEVELOPMENT (TECH SKILLS) INTERESTS Software Development INTERESTS Software Engineering PERSONAL SKILLS Web Development INTERESTS Mobile App Development PERSONAL SKILLS Data Science PERSONAL SKILLS Sports INTERESTS (Basketball INTERESTS , Karate INTERESTS) Music INTERESTS & Dancing B.Sc. Level-2 Software Project Online Shopping App Developed an Online shopping application which is user friendly and takes individual user preferences to provide an optimum user experience for both buyers and sellers. Used technologies are Ionic MOBILE APP DEVELOPMENT (TECH SKILLS) 3, Angular WEB DEVELOPMENT (TECH SKILLS) 7, ASP.NET Core OTHER TECHNOLOGIES and MSSQL DATABASE (TECH SKILLS) Server. B.Sc. Level-1 Hardware Project PROJECT KEYWORDS Automated Snellan Chart Developed a hardware system to automate PROJECT KEYWORDS the process of eye checking which is still done by a medical officer. The technology PROJECT KEYWORDS is based on Micro-Controllers (ATmega32) including a Bluetooth and IR sensor. 2017 WORK AND OTHER EXPERIENCE Participated ACHIEVEMENT TYPE in HackMoral 2.0 COMPETITION / EVENT (2019) Participated

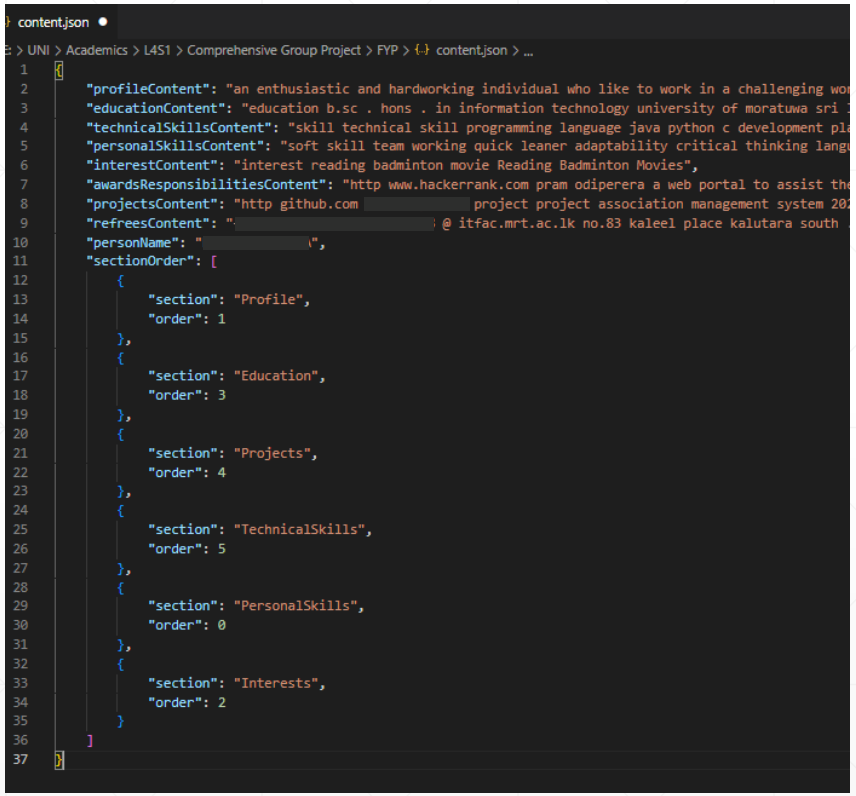
Using NER to identify entities

Implementation

After getting all the content then the system write all the content to a Json file.

```python
# Export content Json
import json

# some JSON:
result = {
"profileContent":profileContent,
"educationContent":educationContent,
"technicalSkillsContent":technicalSkillsContent,
"personalSkillsContent":personalSkillsContent,
"interestContent":interestContent,
"awardsResponsibilitiesContent":awardsResponsibilitiesContent,
"projectsContent":projectsContent,
"refreesContent":refreesContent,
"personName":personName,
"sectionOrder":sectionOrder
}

# parse x:
y = json.dumps(result)

# Writing to sample.json
with open("content.json", "w") as outfile:
    outfile.write(y)
```

Append extracted data and write those content to a Json file



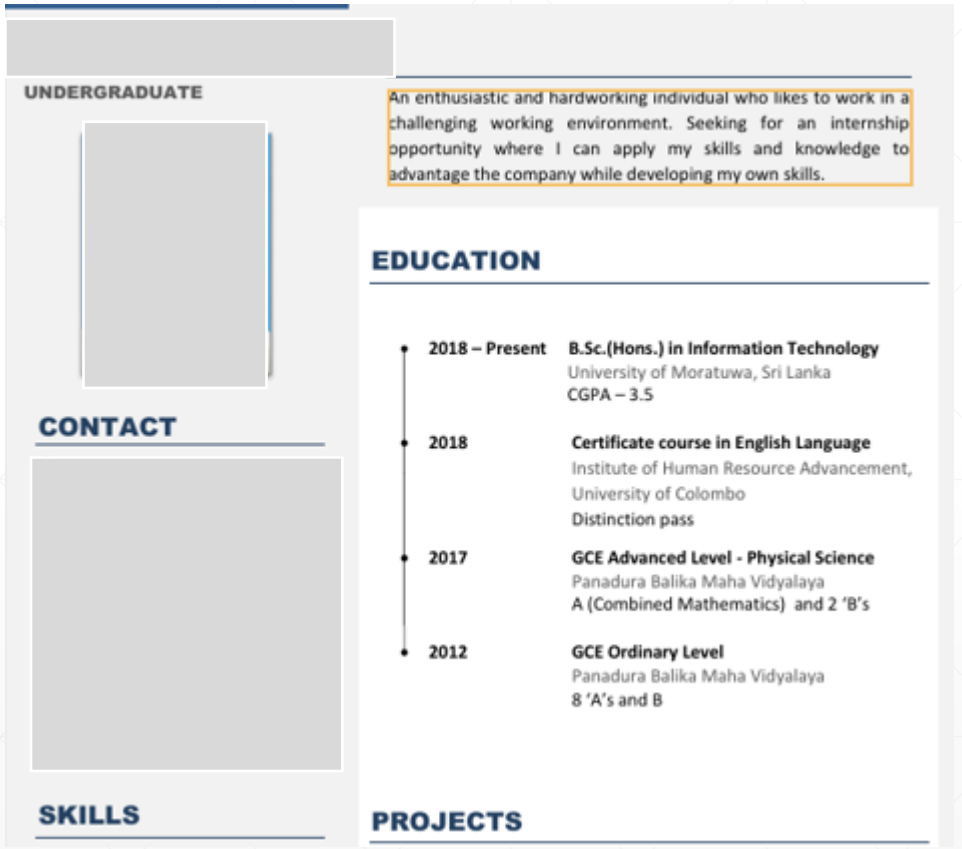
Final Output Json file

Implementation

As the final part of this module, it will generate separate images for each section with the bounding boxes for relevant content

```
def export_section_img(section_box_arr, section_name):
    current_page = 0
    box_to_draw = []
    for box in section_box_arr:
        if current_page == box.page:
            box_to_draw.append(box)
        else:
            plt.figure(figsize=(20,14))
            img = lp.draw_box(pdf_images[current_page], box_to_draw)
#             plt.imshow(img)

            img.save('Section_imgs/temp_'+section_name+'_'+str(current_page)+'.png', "PNG")

            box_to_draw = []
            current_page = box.page
            box_to_draw.append(box)

    plt.figure(figsize=(20,14))
    img = lp.draw_box(pdf_images[current_page], box_to_draw)
#     plt.imshow(img)

    img.save('Section_imgs/temp_'+section_name+'_'+str(current_page)+'.png', "PNG")

export_section_img(profileContentBox, 'profile')
export_section_img(projectsContentBox, 'projects')
export_section_img(educationContentBox, 'education')
export_section_img(technicalSkillsContentBox, 'technical_skills')
export_section_img(personalSkillsContentBox, 'personal_skills')
export_section_img(interestContentBox, 'interests')
export_section_img(awardsResponsibilitiesContentBox, 'awards_responsibilities')
export_section_img(refreesContentBox, 'referees')
```

Generating section wise images



Generated Image for profile section

Evaluation

Evaluation of Multiclass Classification Model for Resume Section Prediction: F1-Score Analysis

The system used the F1-Score assessment metric to evaluate the accuracy with which its multiclass classification model performed in predicting each section of a resume.

Section	SVM (F1-Score)	NaiveBayes (F1-Score)
Profile	0.90	0.80
Education	0.91	0.82
Projects	0.92	0.90
Referees	0.97	0.98
Extra Curricular/ Roles and Responsibilities	0.79	0.76
Awards Achievements	0.85	0.85
Interest	0.88	0.78
Personal Skills	0.88	0.73
Technology/Technical skills	0.94	0.91
Work Experience	0.39	0.00

F1-Scores for both algorithms for each section

SVM with SGDClassifier	0.8881431767337807
Multinomial NaiveBayes	0.8400447427293065

Accuracy of the models

Evaluation

Cosine Similarity Analysis for Section Content Comparison

The cosine similarity metric is used to evaluate how closely the extracted relevant section to the actual section content in order to evaluate the system's accuracy and efficiency.

Section	Actual	Predicted
Profile	An enthusiastic and passionate individual v	an enthusiastic and passionate i
Education	B.Sc(Hons.) in Information Technology Univ	no 230 1st step nawamalkaduwa
Technical Skills	C Java HTML JavaScript ReactJS NodeJS SQL	c java html javascript reactjs noc
Personal Skills	Good communication Team work Leadershi	good communication leadership
Interests	Animal Welfare Hand crafting Volunteering	travelling gaming interest
Awards and Responsibilities	HackX Inter-University Startup Challenge 2(	www.linkedin.com in dini thi ar
Project	INSFRA SMART OFFICE A software system w	http github.com dinzie95 http tv
Referees	Dr. (Mrs.) G. U Ganegoda Senior Lecturer, F	94 71 380 6807 +94 77 728 6807 d
Profile	A committed hard-working third year unde	objective a committed hard wo
Education	UNIVERSITY OF MORATUWA (2017-PRESENT	education overall gpa 3.46 unive
Technical Skills	Programming Languages Java, C Databases	technical skill programming lan
Personal Skills	Project management Complex problem sol	interpersonal skill project mana
Interests	Volunteering Blogging Reading Swimming	interest volunteering blogging r
Awards and Responsibilities	Student Representative (2017-2018) Memb	linkedin linkedin.com in nethm
Project	A SOFTWARE SYSTEM FOR BUSINESS ANALY	project project a web based app
Referees	Dr. G. Upeksha Ganegoda Senior Lecturer D	phone +94 76 993 2442 +94 11 28

Dataset for Cosine similarity analysis

Section	Average Cosine Value
Profile	0.8848024768311141
Education	0.7301637245094814
Projects	0.7836110811709401
Technical Skills	0.7353279814496185
Personal Skills	0.8700204069920932
Interests	0.740757026121876
Awards and Responsibilities	0.7716714248985228
Referees	0.8270679024187793

Average cosine values for each section

Conclusion

- The proposed system has taken on the problem of tackling an important issue in the area of layout-aware text extraction for resumes
- Existing techniques have frequently ignored the crucial element of resume formatting in favor of extracting primary entities from resumes. But the suggested system, on the other hand, is a ground-breaking strategy that carefully considers the format of the resume.
- The proposed system also has limitations when processing resumes with unconventional layouts or intricate formatting.
- The incorporation of deep learning techniques, coupled with access to extensive and diverse datasets, holds immense potential for refining the accuracy and adaptability of the proposed system.

Thank You!