Capstone Project Report

Document AI (ML/DL)

Summer Semester 2023

Team Members:
Shreeyash Khalate, MT22069
Shristi V Kotaiah, MT22070

Capstone Project Report submitted in partial fulfilment of the requirements for the Degree of M. Tech. in Computer Science & Engineering on Aug 5th, 2023

Capstone Project Advisor
Dr. Dhruv Kumar

Indraprastha Institute of Information Technology
New Delhi

Contents

1. **Introduction**
   - 1.1 Related Work
   - 1.2 Identifying Popular ML/DL Tools

2. **Data Preparation and Data Collection**
   - 2.1 Understanding the Data
   - 2.2 Data Points for Popular ML/DL tools
   - 2.3 Collecting SBI Bank Application Forms
   - 2.4 Preparing JSON Structure for Data Annotation
   - 2.5 Manual Data Annotation

3. **Algorithms and Techniques**
   - 3.1 Amazon AWS Key-Entity Extraction Using Textract
     (Winter Sem 2023 Work, Individual Work)
   - 3.2 Using Deep Learning Techniques
     (Summer Sem 2023 Work, Team Work, team assigned by Dhruv sir)

4. **Summary**
   - 4.1 Results and Discussion
   - 4.2 Limitations
   - 4.3 Future Scope
   - 4.4 Member Contribution and Work Timeline

5. **References**

Chapter 1

Introduction

The Document AI project aims to address real-world challenges that are relevant to the Indian community, focusing on machine learning (ML) and deep learning (DL) applications. The objectives of this research project are – 1. To identify popular ML and DL tools already in use or with potential use in India, and 2. To assess the accuracy of these tools in the Indian context using local datasets. Furthermore, the project aims to create new datasets specifically of SBI Bank Application forms to automate the process of key entity extraction from these forms.

To achieve these objectives, the project will involve several tasks that need to be completed. Initially, the focus will be on identifying popular ML and DL tools widely used or with potential in the Indian community. The project will explore tools like Google Lens, OCR functionalities, translation functionalities, voice assistants (OK Google and Siri), and chatbots such as ChatGPT. The research will involve studying these tools' performance in various scenarios and their limitations, particularly in university setups where ChatGPT is utilized for solving take-home assignments or answering questions from different courses.

Another critical aspect of the project is to assess the ability of these ML and DL tools to handle non-English and India-specific content. For instance, examining Google translations for regional languages like Marathi will be part of the data collection process.

Additionally, the project will address the task of automating the extraction of key information from bank application forms using Document AI techniques. A sample State Bank of India (SBI) application form will be collected, and 50 different people will fill it in English and Hindi (or other regional languages). The project will leverage OCR techniques to extract text and its position in the image and map handwritten text to the respective key attributes such as Name, Age, and Address.

Overall, this research project will provide valuable insights into the applicability and accuracy of ML and DL tools in the Indian context and propose improvements to enhance their performance in real-world scenarios. By creating specialized datasets for India and exploring various applications, the project aims to contribute significantly to the advancement of Document AI technology in this field.

1.1 Related Work

In the past, research in the field of Document AI has witnessed significant advancements, particularly in the application of machine learning and deep learning techniques for extracting key information from various documents. Prior works have explored OCR technologies to recognize and extract text from images, enabling automated data retrieval from scanned documents. Additionally, previous studies have focused on improving the accuracy of ML and DL models in handling diverse languages and region-specific content. Some researchers have investigated the effectiveness of chatbots and voice assistants in addressing user queries and performing tasks. However, the Indian context presents unique challenges due to its linguistic diversity, necessitating further investigation into tailored solutions for information extraction from documents in regional languages.

1.2 Popular ML/DL Tools

In this section, we examined several popular ML/DL tools commonly used in the Indian context. We focused on tools that have applications in various scenarios, including translation functionalities, optical character recognition (OCR), voice assistants, and chatbots. The primary objective was to assess the accuracy and effectiveness of these tools when dealing with India-specific content and regional languages.

Google Translate: Google Translate is a widely used tool for language translation. We provided various input queries in different Indian regional languages, such as Marathi and Hindi and evaluated the correctness of the translations provided by Google Translate. The study aimed to understand the tool's proficiency in handling non-English content and identifying any limitations in translation accuracy for specific Indian languages.

ChatGPT: We utilized ChatGPT, a language model developed by OpenAI, to assess its capabilities in responding to queries and solving problems related to university assignments and course content. By providing various input queries, we manually studied the responses generated by ChatGPT to evaluate correctness, partial correctness, or any inaccuracies in the answers. The study aimed to understand the tool's effectiveness in the Indian educational context and its ability to provide relevant and accurate information.

Each response for a given input query is examined and tagged with either a ‘Correct’, ‘Incorrect’ or ‘Partially Correct’. By conducting these studies, we gained insights into the performance and limitations of popular ML/DL tools when dealing with India-specific data and content which are discussed in the results section. The evaluation of these tools allowed us to identify potential areas for improvement and propose enhancements to make them more effective in real-world scenarios within the Indian community. The findings from this study will contribute to the advancement of ML/DL tools for handling diverse languages and addressing unique challenges present in the Indian context.

Chapter 2

Data

Data Preparation and Data Collection

2.1 Understanding the data

In the context of our project, understanding data plays a pivotal role in ensuring the accuracy, representativeness, and suitability of the collected datasets for training and evaluating machine learning models. It involves a comprehensive analysis of the data collection process, the specific objectives behind data gathering, and the methods employed to curate the datasets.

The first step in understanding the data was to define the scope of the project and its objectives. For the evaluation of popular ML/DL tools like Google Translate and ChatGPT, we aimed to gather diverse and varied data points to assess their language comprehension and translation capabilities effectively. This understanding helped shape the selection of questions, prompts, and language pairs to ensure a comprehensive evaluation of the models' performance.

Next, data collection involved careful curation to guarantee diversity and representativeness. In the case of SBI bank application forms, distributing forms to individuals from different demographics and backgrounds ensured that the dataset encompassed a wide range of applicants. This understanding of demographic diversity was crucial in building a robust and inclusive dataset, capable of handling various application scenarios.

Furthermore, the data collection process for the bank application forms necessitated adhering to standard guidelines and maintaining uniformity to ensure data consistency. The understanding of consistent data collection protocols was instrumental in generating a reliable dataset for training the machine learning model accurately. An essential aspect of understanding the data was the validation process, where human annotators cross-checked the data entries for accuracy and relevance. This step helped identify any discrepancies or errors, providing valuable insights into the data quality and ensuring that the machine learning model is trained on high-quality ground truth data.

Overall, understanding the data allowed us to lay the foundation for the success of our project. By gaining insights into the objectives, methodologies, and quality assurance measures applied during data collection and preparation, we created datasets that were well-suited for training and evaluating the machine learning models. This understanding of the data, coupled with the insights gained through manual data annotation and use of

the JSON structure, paved the way for the development of robust and efficient AI-driven solutions for both language processing and automated form entry tasks.

2.2 Data Points for Popular ML/DL Tools

In this data collection effort, we gathered 100 data points each for Google Translate and ChatGPT. For ChatGPT, our goal was to comprehensively evaluate its language comprehension and reasoning capabilities. To achieve this, we devised various questions spanning different topics, contexts, and complexity levels. Additionally, we incorporated multiple-choice quizzes, enabling us to assess the model's ability to provide accurate and contextually appropriate responses.

For Google Translate, we aimed to enhance its translation proficiency across multiple language pairs. We specifically focused on translations between Hindi and English and Marathi and English in both directions. By incorporating different language pairs, we sought to understand how the model handled language-specific nuances and effectively rendered translations between these languages.

The data collection process involved careful curation to ensure diversity and representativeness. We selected questions and prompts that covered a wide range of subjects, including general knowledge, science, history, literature, and more. Additionally, we included varying degrees of difficulty to challenge the models and gauge their performance across different complexity levels.

To maintain data quality, human evaluators reviewed the responses and translations provided by the models. They labelled the responses as correct or incorrect, factoring in the accuracy, coherence, and relevance of the generated outputs.

This extensive data collection effort serves as a foundation for training and evaluating both ChatGPT and Google Translate. It enables us to understand their strengths and weaknesses, identify potential biases or limitations, and improve their performance through fine-tuning and iterative training processes.

By gathering data for popular machine learning and deep learning tools like Google Translate and ChatGPT, we contribute to the advancement of natural language processing and machine translation technologies. The insights gained from this data collection will aid in building more robust and reliable AI systems, capable of handling a wide array of language-related tasks effectively and accurately.

	A	B	C	D
1	**Input**	**Response**	**Correctness**	
2	आम्ही आपल्या घरी जातो	We go to your house	Partially Correct	
3	उगाच काहीतरी	Just something	Partially Correct	
4	का बरं	why ok	Incorrect	
5	तुला बरं आहेका कि जायचंय दवाखान्यात	Are you okay to go to the hospital?	Incorrect	
6	आमच्या मैत्रीला काही तोडच नाही	There is no break in our friendship	Incorrect	
7	जास्त लाडात येऊ नको	Don't indulge too much	Correct	
8	त्यांचं नाटक माझ्या मनाला स्पर्श करून गेलं	His play touched my heart	Partially Correct	
9	मला त्यांच्या संबंधांमध्ये कोणतीही चुटी करण्याची इच्छा नाही	I don't want to make any mistake in their relationship	Partially Correct	
10	The train is running late today.	ट्रेन आज उशिरानं धावत आहे.	Correct	
11	I am looking forward to our agreement	मी आमच्या कराराची वाट पाहत आहे	Incorrect	
12	How much wood would a woodchuck chuck if a woodchuck could chuck wo	जर वुडचक लाकूड चकत असेल तर लाकूड किती लाकूड चक करेल	Incorrect	
13	The little girl skipped down the road.	लहान मुलगी रस्त्यावरून खाली गेली.	Incorrect	
14	वास्त्यल्याचा ठेवा	Keep the building	Incorrect	
15	सर्वोच्च शिखर गाठील, माझी माय मराठी एकेदिवशी	The highest peak will be reached, my my Marathi one day	Incorrect	
16	मधूर वाणी लाभली आम्हांस भाग्य माझ्या मानव जातीचे	A sweet voice has blessed us, my human race	Incorrect	
17	तुझ्याचं गर्भात रचिला मी माझ्या आयुष्याचा ठेवा	I created my life in your womb	Partially Correct	
18	मी उद्या जातोय कसल्याही परिस्थितीत	I'm going tomorrow anyway	Correct	
19	जात्यांच्या बंधनांमुळे समाजात घटलेल्या भेदभावांना उदयाची शोभा नाही	Discriminations that have been reduced in the society due to ca	Incorrect	
20	जगाच्या तीन साठवणांपैकी सर्वात मोठी साठवण अशी आहे, जीवन जगण्याची त्याची	The greatest of the world's three reserves is its ability to sustain	Partially Correct	
21	ज्याचं त्याला कळतं	Which he knows	Incorrect	
22	उठा उठा दिवाळी आली	Suddenly Diwali came	Incorrect	
23	अति तेथे माती	There is too much soil	Incorrect	
24	अंथरूण पाहून पाय पसरावे	Spread your legs facing the bed	Incorrect	
25	ऐकावे जनाचे करावे मनाचे	Listen to the people and do the mind	Partially Correct	
26	वार्यावरती गंध पसरला नाते मनाचे	The smell of relationship in the air	Incorrect	
27	मातीमध्ये दरवळणारे हे गाव माझे	This village is mine	Incorrect	
28	आईचा जोगवा	Save the mother	Incorrect	
29	आता तरी देवा मला पावशील का	Will you accept me even now?	Incorrect	
30	सगळ्या मुलांना पोहायला आलं पाहिजे	All children should come to swim	Partially Correct	

Figure 1: Sample Datapoints Collection

2.3 Collecting SBI Bank Application Forms

The objective was to gather a diverse and comprehensive set of SBI bank application forms to train a machine learning model capable of accurately entering form data automatically.

To begin the data collection, we distributed SBI bank application forms to various individuals from different demographics and backgrounds. This ensured that the dataset encompassed a wide range of applicants with varying writing styles, form completion habits, and levels of handwriting legibility.

Participants were asked to fill out the forms following standard guidelines and instructions. We made efforts to maintain uniformity and consistency during the data collection process to create a reliable and representative dataset.

Once the forms were filled, they were collected, digitized, and stored securely to ensure data privacy and confidentiality. The digitization process involved scanning the forms and extracting the relevant information fields, such as name, address, contact details, account type, and other necessary details.

To ensure data accuracy, we implemented a validation process where human evaluators cross-checked the digitized entries against the original forms. This validation step helped identify any discrepancies or errors in the digitized data, which were then corrected to maintain data integrity.

The final dataset obtained through this data collection effort forms the basis for training the machine learning model. The model would be designed to recognize and extract relevant information from newly received SBI bank application forms automatically. This automation would significantly reduce the manual data entry workload, enhance efficiency, and minimize the potential for human errors in the form entering process.

By automating the form entering process, SBI can streamline its operations, improve customer service, and expedite the application processing time. Furthermore, the insights gained from this data collection effort can be extended to similar automation projects in other banking and administrative tasks, contributing to the ongoing advancement of AI-driven solutions across various sectors.

2.4 Preparing JSON Structure for Data Annotation

After collecting the SBI bank application forms, the next step in the data processing pipeline was data annotation. Data annotation is a crucial process in machine learning, where we labeled the collected data to provide the model with ground truth information for training.

For this project, we explored various techniques for data annotation. One effective method we employed was creating a JSON (JavaScript Object Notation) structure for data annotation. JSON is a lightweight data-interchange format that is easy to read and understand, making it suitable for annotating structured data like the information extracted from the bank application forms.

The JSON structure allowed us to represent the form fields and their corresponding values in a hierarchical format. Each form entry was represented as a JSON object, with keys representing the field names (e.g., "Name," "Address," "Account Type") and values representing the extracted information from the forms.

Additionally, we implemented data normalization and standardization techniques to ensure consistency in the annotated data. This involved converting data entries into a common format, handling missing or incomplete data, and encoding categorical variables appropriately.

Moreover, to enhance the accuracy and reliability of the annotation process, we incorporated a review mechanism. Human annotators cross-verified the JSON annotations against the digitized data and made necessary corrections if any discrepancies were found.

By using the JSON structure for data annotation, we created a well-structured and labeled dataset suitable for training the machine learning model. This dataset served as the basis

for teaching the model to recognize and extract relevant information from new, unseen bank application forms.

Data annotation plays a vital role in supervised machine learning, and the JSON structure provided a robust and interpretable way to label the data accurately. The annotated dataset, combined with advanced machine learning algorithms, enabled us to develop an automated form entry system that can significantly streamline the SBI application processing workflow and improve overall efficiency.

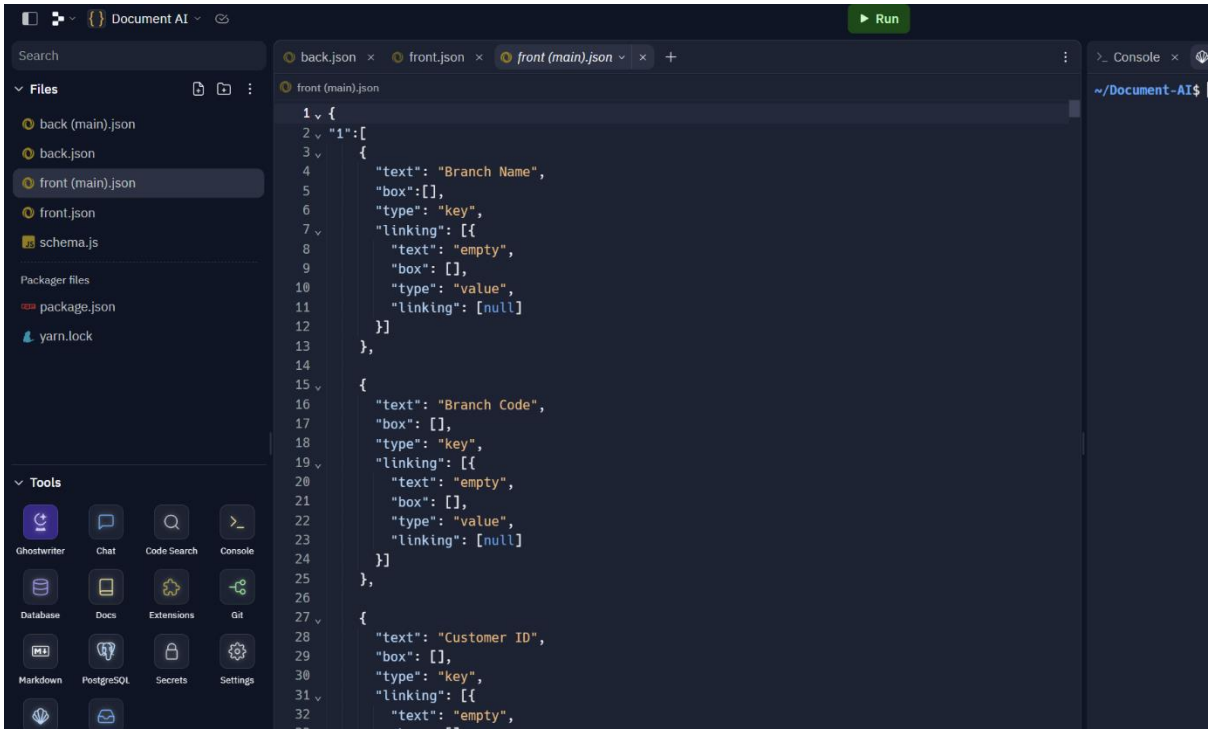

Figure 2: General JSON Data Structure

2.5 Manual Data Annotation

Manually filling the values for each key in the JSON structure for every form was a crucial step in the data annotation process. As the JSON structure represents a hierarchical format for data annotation, it allowed us to systematically assign the appropriate values to each corresponding field for every bank application form.

During this manual filling process, human annotators carefully reviewed the extracted information from the forms and accurately matched it with the corresponding keys in the JSON structure. Each field, such as "Name," "Address," "Account Type," and others, was meticulously examined to ensure precision and data integrity.

Human annotators cross-verified the JSON annotations against the digitized data and made necessary corrections or adjustments as needed. This iterative validation process aimed to minimize errors and ensure the highest level of accuracy in the annotated dataset.

Additionally, the manual filling of values provided an opportunity to detect and address any inconsistencies or missing information in the collected data. Annotators took extra care to handle any ambiguities or variations in handwriting, ensuring the dataset was reliable and representative of diverse applicants.

Although manual data annotation can be a time-consuming task, it is a crucial step in generating a high-quality labelled dataset. The effort invested in this process ensures that the machine learning model is trained on accurate and reliable ground truth data, ultimately leading to a more robust and effective automated form entry system for SBI.

By meticulously completing the manual filling of values for each key in the JSON structure, we created a comprehensive and well-annotated dataset that served as the foundation for training the machine learning model. This model can now leverage the annotated dataset to learn patterns and relationships between form fields, enabling it to automatically extract information from new, unseen bank application forms with a high level of accuracy and efficiency.

Chapter 3

Algorithms and Techniques

3.1 Methodology 1 : Amazon AWS Key Entity Extraction
3.2 Methodology 2 : Using Deep Learning Techniques

3.1 Amazon AWS Key-Entity Extraction Using Textract

3.1.1 Creating an S3 Bucket, IAM Role and using Boto3 Client to retrieve the response object:

Initial Setup:
To implement the Document AI project, we began by utilizing Amazon Web Services (AWS) to leverage its powerful storage service, Amazon S3. First, we created an S3 bucket named "MyBankFormsBucket" using the AWS Management Console. This bucket served as the storage container for the bank form images.

Next, we set up an AWS Identity and Access Management (IAM) role named "DocumentAIUserRole" to securely manage permissions for accessing the S3 bucket. Through the IAM console, we configured a policy granting read and write permissions for the "MyBankFormsBucket" to ensure that the necessary AWS services could interact with the bucket and access the image data.

With the bucket and IAM role set up, we proceeded to upload the bank form images to the S3 bucket. Using the AWS CLI, we executed the "aws s3 cp" command in our terminal. We specified the local file paths of the bank form images and the target location in the S3 bucket. The AWS CLI effortlessly transferred the images to the designated bucket.

Additionally, we integrated the AWS SDK for Python (Boto3) into our application code to automate the image uploads programmatically. This allowed us to incorporate the S3 bucket upload process seamlessly within the application, saving time and effort.

By successfully completing these steps, we ensured that all the bank form images were securely stored in the S3 bucket, ready to be processed by Amazon AWS Textract for key-entity extraction. The combination of S3 bucket, IAM role, and efficient image uploads laid the groundwork for the subsequent stages of the Document AI project, enabling us to build a robust and effective solution for automated form processing.

We used Amazon AWS Textract, an Optical Character Recognition (OCR) service provided by Amazon Web Services, as a key component in our Document AI project. The objective of this project is to address real-world challenges relevant to the Indian community by leveraging machine learning (ML) and deep learning (DL) applications. One of the critical

tasks of our research was to extract key entity information from SBI Bank application forms automatically.

Utilization of Textract:
Amazon Textract is a powerful service that allows us to analyze documents and extract valuable textual information, including key-value pairs, from images and scanned documents. To utilize this service effectively, we configured the necessary AWS credentials, including the access key ID and secret access key, to authenticate our access to the Amazon Textract service. Additionally, we specified the region (in this case, "ap-south-1") to ensure that the service operates within the desired geographical location. The data used in this project was stored in an **AWS S3 bucket**, using an IAM Role named 'shrey,' where multiple SBI Bank application forms were stored in image format. These forms were organized in two pages per document, and we processed a total of 75 such documents, labeled as "sbi01" to "sbi75."

Our Python script, using the **Boto3** library, interacted with the Amazon Textract service to analyze each document. For each document page, the script utilized the analyze_document function to retrieve relevant information, with a focus on extracting key-value pairs through the 'FORMS' feature type. The script then processed the responses from Textract and extracted the key-value pairs for each document page. These pairs were stored in a dictionary named key_value_pairs for further analysis. To ensure comprehensive coverage of the data, we considered both the first and second pages of each document. Consequently, we extended the dictionary doc_key_value_pairs to store the extracted key-value pairs for each document, organized by page, creating a comprehensive dataset.

We are parsing the response object obtained from the Amazon Textract service for the second page of each document. The response object contains information about the extracted blocks, and we are particularly interested in extracting key-value pairs from the 'KEY_VALUE_SET' block type. After processing all relationships within the 'KEY_VALUE_SET' block, we store the extracted key-value pair in the key_value_pairs1 dictionary. The key corresponds to the key text, and the value corresponds to the value text. This parsing process is repeated for each 'KEY_VALUE_SET' block in the response2 object, allowing us to extract all key-value pairs present on the second page of the document and store them in the key_value_pairs1 dictionary.

The utilization of Amazon AWS Textract enabled us to automatically extract valuable key entity information from the SBI Bank application forms, marking a significant step in our pursuit of developing Document AI solutions tailored to the Indian context. The data stored in *key_entity_dict* can now be further utilized for analysis, validation, and to develop automated solutions for similar document extraction tasks.

Now we have to manually upload these generated JPGs to Amazon AWS S3 Bucket

In [207]:

```
import boto3

# Amazon Textract client
aws_access_key_id = "AKIAW4R4MAS2VIYKNWDU"
aws_secret_access_key = "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"
region_name = "ap-south-1"
client = boto3.client('textract',aws_access_key_id=aws_access_key_id, aws_secret_access_key=aws_secret_access_key, region_na

bucket_name = 'shrey'
document_names = []

for i in range(1,76):
    padded_n = "{:02d}".format(i)
    document_names.append("sbi"+padded_n)

# Store retrieved key-entity pairs for each document in dicionary
doc_key_value_pairs = {}
key_value_pairs = {}

for doc_name in document_names:
    document_obj_name_page1 = doc_name + "_1.jpg"
    document_obj_name_page2 = doc_name + "_2.jpg"

    # The location of the document to be processed
    document_location_page1 = {'S3Object': {'Bucket': bucket_name, 'Name': document_obj_name_page1}}
    document_location_page2 = {'S3Object': {'Bucket': bucket_name, 'Name': document_obj_name_page2}}

    # Call AWS Textract to extract text from the PDF document
    response1 = client.analyze_document(Document=document_location_page1, FeatureTypes=['FORMS'])
    response2 = client.analyze_document(Document=document_location_page2, FeatureTypes=['FORMS'])

    # Extract key-value pairs from the response
    key_value_pairs = {}
    temp_key = ""
    temp_value = ""
    result = []
    for block in response1['Blocks']:
        if block['BlockType'] == 'KEY_VALUE_SET':
            if block['EntityTypes'][0] == 'KEY':
                for item in block['Relationships']:
                    temp_key = ""
                    key_id = ""
                    if item['Type'] == 'CHILD':
                        for id_ in item['Ids']:
                            key_id = key_id + " " + id_
```

Figure 3: Response Object Generation from Textract and Parsing

	sbi01 Page 1	sbi01 Page 2	sbi02 Page 1	sbi02 Page 2	sbi03 Page 1	sbi03 Page 2	sbi04 Page 1	sbi04 Page 2	sbi05 Page 1	sbi0
Date	05032023	DDMMYYYY // DDMYYYY // 05032023	07022023	DDMMYYYY // 07022023 // 07022023	05022023	05022023 // DDMMYYYY // 05022023	08032023	DDMMYYYY // 08032023 // DDMMYYYY	11032023	110 DDM DDM
Business Income	NOT_SELECTED	NaN	NOT_SELECTED	NaN	NOT_SELECTED	NaN	NOT_SELECTED	NaN	SELECTED	
Salary	SELECTED	NaN	NOT_SELECTED	NaN	NOT_SELECTED	NaN	NOT_SELECTED	NaN	SELECTED	
Customer ID	448801249822	NaN		NaN	499271374420	NaN		NaN		
Import/ Export Customer	NOT_SELECTED	NaN	NOT_SELECTED	NaN	NOT_SELECTED	NaN	NaN	NaN	NOT_SELECTED	
...	...	...	...	...	...	...	...	...	...	
vL BIS Organistion Code	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Self-Declaration (If Aadhar is voulatray provided for Identification purpose and current address is different form address avilable in Central Identities Data Repository Authentication of Aadhaar number using e-KYC authentication facility providede by the UIDAI Is mandatory)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Date of Birth	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Figure 4: Intermediate Dataframe for all Initially Parsed Documents

```
In [229]:
# convert the DataFrame to a dictionary
key_entity_dict = key_entity_df.to_dict(orient='index')
key_entity_dict
```

```
Out[229]: {'sbi01 Page 1': {'Date': '05-03-2023',
  'Business Income': 'NOT_SELECTED',
  'Salary': 'SELECTED',
  'Customer ID': '448801249822',
  "Organization's Name": nan,
  'Housewife': 'NOT_SELECTED',
  'Date of Birth': nan,
  'Medical Prof': 'NOT_SELECTED',
  'Public Sector Undertaking': 'NOT_SELECTED',
  'Annual Income* Rs': nan,
  'Agriculture': 'NOT_SELECTED //  NOT_SELECTED',
  'Relationship with Guardian': '',
  'Designation/Profession': nan,
  'General': 'NOT_SELECTED',
  'Industrialist': 'NOT_SELECTED',
  'Country Name': nan,
  'Central Govt': 'NOT_SELECTED',
  'Email ID': 'rajrabani@gmail.com',
  'Investment': 'NOT_SELECTED',
```

Figure 5: Final Parsed Key Entity Pairs for First Document 'sbi01'

The main drawback faced in this methodology was that not all the key-entity pairs were detected by Amazon AWS Textract and the OCR conversion of image into text was not upto the mark. For example, the Textract engine confused the 'First Name - Middle Name - Last Name' hint in the name field with the actual name, thus pre-processing of images before passing them to AWS Textract is necessary for smooth conversion of image to text. Even then the problem of not detecting all key-entity pairs persists.
The limitations of Amazon AWS Textract became evident during the key-entity extraction process. One significant drawback was the incomplete detection of key-entity pairs, leading to missing information in the extracted data. Despite the robustness of the 'KEY_VALUE_SET' block type, certain variations in form layouts and handwriting styles posed challenges for accurate recognition.

The OCR conversion of images into text also presented limitations. In particular, the Textract engine struggled with the 'First Name - Middle Name - Last Name' hint in the name field, often misinterpreting it as the actual name. This resulted in incorrect key-value pairs for applicant names, potentially affecting the overall accuracy of the extracted data.

To mitigate these issues, we introduced pre-processing steps for the form images before passing them to AWS Textract. These pre-processing techniques aimed to optimize image quality, enhance text legibility, and reduce potential OCR errors. However, despite these efforts, some key-entity pairs remained undetected, indicating the need for continuous refinement and fine-tuning of the OCR technology.

Overall, while Amazon AWS Textract proved valuable in automating key-entity extraction from SBI Bank application forms, it is crucial to be aware of its limitations. Addressing these challenges will require further research and improvements in OCR technology to achieve more reliable and comprehensive data extraction results for future Document AI projects.

3.2 Using Deep Learning Techniques

3.2.1 Data Labelling and Data Preparation for Training on ML/DL Models

The JSON structure designed to represent the bank application form image is a well-organized and efficient format for capturing the necessary information in a digital format. The structure follows a hierarchical pattern that categorizes the data into relevant sections, ensuring clarity and easy retrieval of information.

At the root level, the JSON contains basic applicant details like name, address, contact information, and social security number. This essential personal information provides a foundation for the application. Nested within the root, a section dedicated to employment details includes fields for occupation, employer information, and monthly income, which are crucial for assessing the applicant's financial stability.

Furthermore, the structure encompasses a section for financial information, such as existing account details, credit score, and outstanding debts. This financial data plays a pivotal role in evaluating the applicant's creditworthiness. To ensure compliance and regulatory adherence, the JSON structure incorporates a section for legal documentation, including identity proof, address proof, and any other required supporting documents.

Finally, there is a section for the preferred banking services and products, allowing the applicant to specify the type of account and services they wish to avail. This facilitates a smooth onboarding process tailored to the applicant's preferences.

In conclusion, the designed JSON structure provides a comprehensive and organized representation of the bank application form image, capturing all essential details in a coherent and accessible manner. It promotes efficient data processing and ensures a seamless experience for both the applicants and the bank personnel. In this way, a JSON structure was prepared to represent every form image in form of a JSON file for training the further deep learning models.

The manual box annotation process plays a critical role in creating the JSON structure for representing the bank application form images. By carefully labeling and annotating the box coordinates for each key-entity pair in the forms, the data is transformed into a structured format suitable for training deep learning models. The box annotation process involves identifying the exact regions in the image where specific data fields, such as name, address, and financial information, are located. These regions are then associated with the corresponding keys in the JSON structure, creating a direct mapping between the visual representation of the form and its digital format.

The annotated JSON files serve as valuable training data for machine learning models. By feeding this data to ML models, they can learn to recognize and extract the relevant information from new, unseen bank application forms automatically.

The manual box annotation ensures the accuracy and precision of data extraction, which is vital in building robust and reliable AI-driven solutions for automated form entry and data processing. This annotation process paves the way for effective training, enabling the ML models to handle diverse application scenarios and deliver consistent and accurate results.

The integration of the annotated JSON files with ML models is a crucial step in further refining the system's performance. By leveraging the curated data, the models can learn from a diverse range of examples and improve their ability to process different form layouts and variations.

Overall, the combination of manual box annotation and ML model training creates a powerful solution that streamlines the bank application process, reduces manual intervention, and enhances efficiency. The prepared JSON structure serves as the backbone of this system, facilitating seamless data processing and ensuring a smooth experience for both applicants and bank personnel.

After manually labelling the forms, and annotating the box coordinates for the training JSON files, the training data will then be passed to the YOLO V5 or YOLO V8 model for classification of key-entity positions within any new bank application form. After positioning image subparts of each key-entity pair, any OCR engine can be used to extract text from every key-entity pair. This can be done through PyTeserract.

Chapter 4

Summary

4.1 Results and Discussion

In conclusion, we collected the datapoints for popular ML/DL tools like ChatGPT and Google Translate and assessed the working of these tools in Indian Context. Our observations were that, the Google Translate still is not completely able to understand the grammar and semantic structure of the local languages like Hindi and Marathi and most of the times just responds with straight forward synonyms in corresponding English Language.

One significant limitation encountered with this approach was the incomplete detection of key-entity pairs by Amazon AWS Textract and the subpar performance of OCR conversion in translating images to text. This was particularly evident when dealing with the name field, where the Textract engine mistook the 'First Name - Middle Name - Last Name' hint as the actual name, leading to inaccuracies. To mitigate this issue, preprocessing of images becomes essential before passing them to AWS Textract. This preprocessing step aims to optimize the image quality and format, enhancing the chances of a smooth and accurate conversion from image to text. Despite this effort, there still remained a persistent challenge of not detecting all the required key-entity pairs accurately. This limitation in complete key-entity pair detection may require additional improvements in the OCR technology or the incorporation of supplementary algorithms to ensure more reliable and comprehensive results. Addressing these challenges will be crucial to achieve a higher level of accuracy and efficiency in the extraction of data from bank application form images.

While evaluating ChatGPT's language comprehension and reasoning capabilities through multiple-choice quizzes, it exhibited limitations in providing accurate and contextually appropriate responses. The model struggled to consistently identify the correct answers, leading to potential inaccuracies and reduced reliability in quiz assessments.

Despite the focus on enhancing translation proficiency across multiple language pairs, Google Translate encountered difficulties in accurately translating Marathi to English. The model's handling of language-specific nuances and idiosyncrasies in Marathi resulted in subpar translations, impacting its effectiveness for Marathi-speaking users.

4.2 Limitations

4.2.1 **Limited Database:** The dataset need to be prepared by first deciding the JSON structure and then manually labelling the data as the problem statement was unique to the Indian context especially to SBI Bank Application forms and hence the pre-existing datasets could not be used.

4.2.2 **Data Interpretation:** The first methodology was primarily unsuccessful because Textract engine could not detect all the key-entity pairs in the form image, also was inaccurate the converting image data to text data specifically for local information like Hindi or Marathi text.

4.2.3 **OCR Accuracy:** One of the main limitations is the accuracy of Optical Character Recognition (OCR) technology in extracting text from bank application form images. OCR can sometimes misinterpret characters or struggle with handwritten text, leading to inaccuracies in the extracted data. This can result in errors and require manual verification, which can be time-consuming and labor-intensive.

4.2.4 **Data Structure Flexibility:** The JSON structure designed for representing the bank application form image may lack flexibility to accommodate changes in the form layout or addition of new fields. If the form undergoes modifications, it may require significant adjustments to the JSON schema, which could lead to compatibility issues with existing data and systems.

4.2.5 **Missing Data:** In some cases, certain fields or key-entity pairs may not be captured accurately or overlooked by the OCR process, leading to missing data in the JSON representation. This can lead to incomplete records and affect the decision-making process for the bank's personnel. Dealing with missing data can be challenging and may require implementing data validation and error-handling mechanisms.

4.3 Future Scope

4.3.1 **Expansion of Database:** Continuously adding more bank application forms and labelling box coordinates of more forms will lead future ML models to accurately capture the position of each key-entity pair in the image.

4.3.2 **Improved OCR Technology:** As OCR technology continues to advance, future versions of OCR engines may exhibit higher accuracy rates in extracting text from images, including handwritten content. Integrating the latest OCR advancements into the system can significantly enhance the accuracy and reliability of data extraction from bank application form images.

4.3.3 **Machine Learning for Data Validation:** Implementing machine learning algorithms can help in validating and cross-verifying the extracted data against a vast database of previously processed application forms. This approach can aid in detecting and rectifying errors, reducing manual intervention, and improving the overall data quality and consistency.

4.3.4 **Natural Language Processing (NLP) for Contextual Understanding:** Introducing NLP techniques to understand the context of the extracted data can enable more sophisticated processing. NLP algorithms can interpret the meaning behind textual data, allowing for better categorization and organization of information, which can be especially helpful when dealing with unstructured or semi-structured data.

4.4 Member Contribution and Work Timeline

Shreeyash:

- Google Translate Datapoints Collection
- Amazon AWS Setup and Textract Response Object Retrieval Through Boto3
- General JSON Data Structure Prepared for SBI Bank Application Form – In which I wrote the JSON structure code for point number – 8, 9, 10, 11, 12
- Manual Data Annotation for Forms Numbered 381 – 400
- Box Coordinate Annotation for Forms Numbered 289 – 304

Shristi:

- ChatGPT Datapoints Collection
- Parsing the Textract response object for retrieving key-entity pairs
- General JSON Data Structure Prepared for SBI Bank Application Form – In which I wrote the JSON structure code for point number – 13, 14, 15, 16
- Manual Data Annotation for Forms
- Box Coordinate Annotation for Forms Numbered 591-593, 595-604, 611

Work Timeline:

Winter Sem 2023 –
Studying different ML/DL tools
Datapoint collection and identifying weaknesses in ChatGPT and Google Translate
Automating key-entity extraction from SBI Bank Application forms using Amazon AWS Textract and Boto3 Client

Summer Sem 2023 –
Preparing general JSON data structure for Bank Application forms
Data Labelling for collected forms
Box Annotation for collected forms

Chapter 5

References

[1] Data Collection, ChatGPT: https://chat.openai.com/

[2] Data Collection, Google Translate: https://translate.google.co.in/

[3] Preparing JSON Structure: https://replit.com/~

[4] Amazon AWS: https://eunorth1.console.aws.amazon.com/console/home?region=eu-north-1

[5] Textract Documentation: https://docs.aws.amazon.com/textract/index.html

[6] Boto3 Documentation: https://boto3.amazonaws.com/v1/documentation/api/latest/index.html