Extracting Structured Data from Unstructured Text

A Project Report
submitted by

Rishabh Garg (DA24C026)

Under the guidance of
PROF. DR. SIVARAM AMBIKASARAN, IIT MADRAS
DR. SRI VALLABHA DEEVI, TIGER ANALYTICS

DEPARTMENT OF DATA SCIENCE AND AI
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
January, 2025

Abstract

This report outlines the development of a pipeline for extracting structured data from unstructured text, integrating technologies such as YOLOv8 for bounding box creation, Optical Character Recognition (OCR), Named Entity Recognition (NER), and a Question-Answering (QA) system. The pipeline employs custom-trained YOLO models, annotated using CVAT, to detect and localise key elements in complex document layouts, which are then processed by OCR to extract textual content. NER, powered by domain-specific models like SciBERT and RoBERTa, identifies and classifies entities such as people, organisations, and locations, organising the raw text into structured information. Additionally, a QA system allows users to query the processed documents for specific information, leveraging the contextual power of the NER models to provide accurate answers. This integrated approach significantly advances document digitisation, information extraction, and query-based retrieval, offering a robust solution for processing complex documents across various domains, including legal, academic, and business contexts.

Acknowledgements

I would like to express my sincere gratitude to my project guides, **Prof. Dr. Sivaram Ambikasaran** and **Dr. Sri Vallabha Deevi**, for their invaluable guidance and support throughout the project. Their expertise and insights have been instrumental in shaping this work. I would also like to thank the faculty members and colleagues at IIT Madras for their encouragement and constructive feedback. Finally, I am grateful to my family and friends for their continuous support and motivation.

Contents

Chapter 1

Introduction

Unstructured text, such as research papers, reports, and legal documents, often contains valuable information that is difficult to process due to its format and complexity. Extracting structured data from such sources is essential for enabling advanced analysis, decision-making, and efficient information retrieval. However, the lack of consistent structure in these documents presents significant challenges.

This project aims to address these challenges by developing an end-to-end pipeline. The system utilises YOLOv8 for bounding box detection, Tesseract OCR for text recognition, and advanced NLP models such as SciBERT for domain-specific understanding. This combination enables the extraction of structured information even from highly complex document layouts, including multi-column research papers.

1.1 Problem Statement

The key objectives of this project are:

- Training a YOLOv8 model for detecting bounding boxes around text regions in research papers.
- Extracting and preprocessing text from detected regions using OCR and cleaning techniques.
- Performing Named Entity Recognition to extract meaningful entities such as names, dates, and terms.
- Enabling efficient information retrieval through context-aware question-answering models.

Chapter 2

Approach

2.1 Pipeline Overview

The pipeline consists of six main stages:

1. **Training YOLOv8 for Bounding Box Detection**: Training a YOLOv8 model on a custom dataset to identify text regions in unstructured documents.
2. **OCR**: Detected regions are processed using Tesseract to extract text, with a focus on handling multi-column layouts.
3. **Text Preprocessing**: Extracted text is cleaned and structured for further analysis.
4. **Named Entity Recognition**: Key entities are identified using SciBERT and SpaCy models.
5. **Context Generation**: Relevant sentences are retrieved based on semantic similarity using SentenceTransformers.
6. **Question-Answering**: Queries are answered using fine-tuned QA models like RoBERTa.

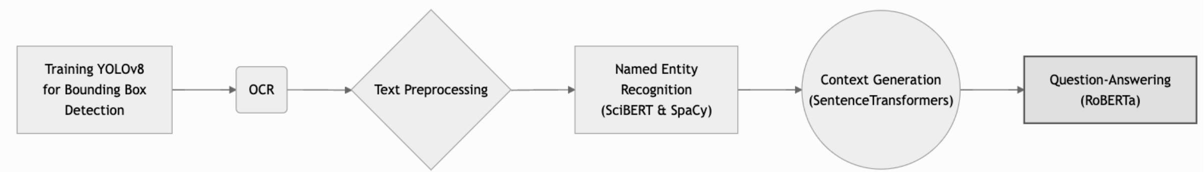

Training YOLOv8 for Bounding Box Detection
OCR
Text Preprocessing
Named Entity Recognition (SciBERT & SpaCy)
Context Generation (SentenceTransformers)
Question-Answering (RoBERTa)

Figure 2.1: Flowchart of the proposed pipeline.

2.2 Training YOLOv8 for Bounding Box Detection

The first step in the pipeline involves training a YOLOv8 model to detect text regions within unstructured documents. YOLOv8 was chosen for its speed and accuracy in object detection tasks. To achieve this, a custom dataset was created using CVAT (Computer Vision Annotation Tool). The annotation process involved:

- Manually labelling text regions in various document types, such as single-column and two-column layouts.

- Including a diverse range of documents, such as research papers, reports, and scanned PDFs, to ensure robustness.
- Splitting the annotated dataset into training and validation sets to prevent overfitting.

The YOLOv8 model was trained using these annotations, fine-tuning its weights to detect text blocks accurately. The training process also involved techniques such as data augmentation to improve the model's generalisation capability. This step ensures that the model can handle various document layouts and complexities effectively.

2.3 Optical Character Recognition

OCR was performed using Tesseract, an open-source tool known for its high accuracy. To address the challenges posed by complex layouts, a page-wise approach was employed, processing each column separately. Additionally, preprocessing steps, such as noise removal and deskewing, were implemented to optimise text recognition. This method proved effective in handling multi-column research papers and other structured documents.

2.4 Named Entity Recognition

NER was conducted using SpaCy's `en_core_web_trf` model and SciBERT, a transformer-based model fine-tuned on scientific literature. These models identified entities such as names, dates, and technical terms, providing structure to the extracted text. SciBERT's domain-specific training enabled it to outperform generic models in recognising entities within research-oriented documents.

2.5 Context Generation and Similarity Computation

The text extracted through OCR was divided into sentences and embedded using SentenceTransformers' `all-MiniLM-L6-v2` model. User queries were similarly embedded, and cosine similarity was calculated to identify the most relevant sentences. The top sentences were concatenated to form a context, which was then used for answering queries. This approach ensured that only the most pertinent information was included in the final context.

2.6 Question-Answering

The QA module leveraged RoBERTa, fine-tuned on SQuAD2.0, to answer queries based on the generated context. The system demonstrated high accuracy for direct questions but faced challenges with multi-sentence answers, highlighting the need for further fine-tuning.

Chapter 3

Implementation Details

3.1 Code Example for Context Generation and QA

In this section, we provide a code example for generating context from unstructured text and performing question-answering (QA) based on the identified context. This process involves the use of **Sentence Transformers** for context generation and a **RoBERTa-based QA** pipeline for answering questions. The code leverages pre-trained models to calculate sentence embeddings, identify the most relevant sentence in the text, and then answer a user query based on that context. The following Python code demonstrates this process:

Listing 3.1: Context Generation and Question-Answering

```
from sentence_transformers import SentenceTransformer, util
from transformers import pipeline

# Load models
sentence_model = SentenceTransformer(’all-MiniLM-L6-v2’)
qa_pipeline = pipeline("question-answering", model="deepset/roberta-
    base-squad2")

# Example text and query
text = "The␣exponential␣growth␣of␣unstructured␣textual␣data␣across␣
    diverse␣domains␣presents␣significant␣challenges␣in␣information␣
    retrieval␣and␣data␣organization.␣This␣study␣explores␣advanced␣
    techniques␣for␣extracting␣structured␣data␣from␣unstructured␣text␣
    using␣a␣combination␣of␣natural␣language␣processing␣(NLP)␣and␣machine
    ␣learning␣approaches.␣We␣propose␣a␣novel␣framework␣that␣integrates␣
    named␣entity␣recognition␣(NER),␣dependency␣parsing,␣and␣deep␣
    learning␣models␣to␣accurately␣identify␣and␣classify␣key␣entities␣
    within␣unstructured␣corpora."

question = "What␣are␣the␣key␣components␣of␣the␣proposed␣framework␣for␣
    extracting␣structured␣data␣from␣unstructured␣text,␣and␣how␣do␣they␣
    contribute␣to␣improving␣extraction␣accuracy?"

# Split text into sentences and encode
sentences = text.split(’.␣’)
sentence_embeddings = sentence_model.encode(sentences,
    convert_to_tensor=True)
query_embedding = sentence_model.encode(question, convert_to_tensor=
    True)
```

```
# Compute cosine similarity
cosine_scores = util.cos_sim(query_embedding, sentence_embeddings)

# Select top sentence
top_sentence = sentences[cosine_scores.argmax()]
context = "␣".join([top_sentence])

# Perform question-answering
result = qa_pipeline(question=question, context=context)
print("Answer:", result['answer'])
```

This example uses two main components: context generation via sentence embeddings and question-answering with a pre-trained model. First, we load the **SentenceTransformer** model, which is capable of encoding sentences into dense vector representations (embeddings). These embeddings are compared using cosine similarity to identify the most relevant sentence that answers the user query. Once the relevant sentence (context) is determined, the **RoBERTa-based QA model** is used to extract the answer to the user's question from that context. The key technologies in this process include:

- **Sentence Transformers**: This library provides a simple API for generating sentence embeddings, allowing the system to understand semantic relationships between sentences and the user's question.
- **RoBERTa-based QA model**: The QA pipeline is built upon the **deepset/roberta-base-squad2** model, which is fine-tuned on the **SQuAD 2.0** dataset. This enables the system to provide precise and contextually relevant answers to questions based on the given document.

3.2 Tools and Libraries

The implementation of the pipeline utilises several advanced tools and libraries, each fulfilling specific tasks to handle the different stages of document processing. Below are the primary tools and libraries employed in this project:

- **CVAT for Annotation and YOLOv8 for Object Detection:** CVAT (Computer Vision Annotation Tool) is used to manually annotate objects in document images, providing accurate ground truth data for training the YOLOv8 model. YOLOv8 (You Only Look Once, version 8) is an object detection model that excels at real-time object localisation within images, enabling the pipeline to automatically detect textual and graphical elements (such as tables, figures, and text blocks) within complex document layouts.
- **Tesseract for OCR: Tesseract** is an open-source optical character recognition engine that is used to extract text from images. Once the bounding boxes for text regions are identified by YOLOv8, Tesseract is employed to recognise the actual characters in those regions. This combination allows for efficient extraction of textual content from scanned documents and images.
- **SciBERT and RoBERTa for NER and QA: SciBERT** is a variant of BERT (Bidirectional Encoder Representations from Transformers) specifically pre-trained

on scientific literature, making it highly effective at recognising domain-specific entities and jargon in technical texts. **RoBERTa**, a robustly optimised version of BERT, is used for the QA task. RoBERTa has been fine-tuned on SQuAD 2.0, a large dataset for question-answering tasks, which enables it to answer specific queries based on context derived from the documents.

- **SentenceTransformers for Embedding-Based Similarity: SentenceTransformers** is a library that extends the functionality of BERT-like models for sentence embeddings, enabling the conversion of sentences into dense vector representations. These embeddings are then compared using cosine similarity to find the most relevant context for a given query. This library simplifies the process of working with text at the sentence level and allows for more accurate context extraction in QA systems.

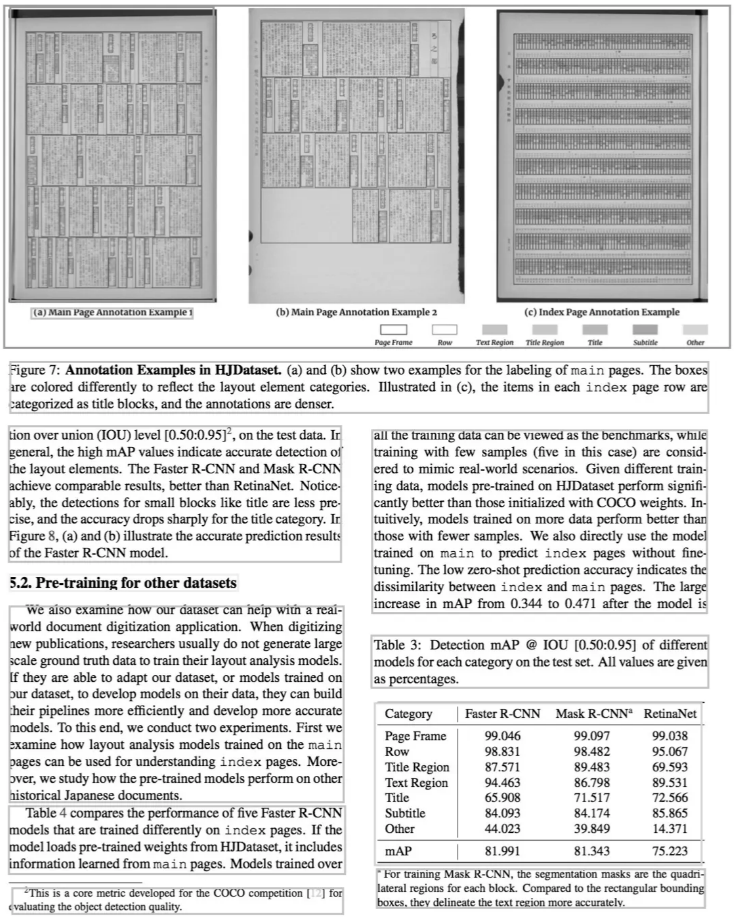
Each of these tools plays a critical role in addressing different challenges inherent in processing unstructured text. The integration of object detection, OCR, NER, and QA models within a unified pipeline ensures that diverse document layouts and content types can be processed effectively, converting unstructured data into a format that is both machine-readable and queryable.

The pipeline's modularity allows for flexible updates and customisation. For instance, the object detection model can be retrained for domain-specific documents, while the QA system can be fine-tuned with additional datasets for enhanced accuracy. The overall system provides a robust foundation for applications across a wide range of domains, including legal document analysis, scientific research, and enterprise content management.

Chapter 4

Results

In this section, we present the key results from the implementation of the pipeline, highlighting the performance of each component, including object detection, OCR, named entity recognition (NER), and question-answering (QA).



(a) Main Page Annotation Example 1 (b) Main Page Annotation Example 2 (c) Index Page Annotation Example

Figure 7: **Annotation Examples in HJDataset.** (a) and (b) show two examples for the labeling of `main` pages. The boxes are colored differently to reflect the layout element categories. Illustrated in (c), the items in each `index` page row are categorized as title blocks, and the annotations are denser.

tion over union (IOU) level [0.50:0.95][2], on the test data. In general, the high mAP values indicate accurate detection of the layout elements. The Faster R-CNN and Mask R-CNN achieve comparable results, better than RetinaNet. Noticeably, the detections for small blocks like title are less precise, and the accuracy drops sharply for the title category. In Figure 8, (a) and (b) illustrate the accurate prediction results of the Faster R-CNN model.

5.2. Pre-training for other datasets

We also examine how our dataset can help with a real-world document digitization application. When digitizing new publications, researchers usually do not generate large scale ground truth data to train their layout analysis models. If they are able to adapt our dataset, or models trained on our dataset, to develop models on their data, they can build their pipelines more efficiently and develop more accurate models. To this end, we conduct two experiments. First we examine how layout analysis models trained on the `main` pages can be used for understanding `index` pages. Moreover, we study how the pre-trained models perform on other historical Japanese documents.

Table 4 compares the performance of five Faster R-CNN models that are trained differently on `index` pages. If the model loads pre-trained weights from HJDataset, it includes information learned from `main` pages. Models trained over all the training data can be viewed as the benchmarks, while training with few samples (five in this case) are considered to mimic real-world scenarios. Given different training data, models pre-trained on HJDataset perform significantly better than those initialized with COCO weights. Intuitively, models trained on more data perform better than those with fewer samples. We also directly use the model trained on `main` to predict `index` pages without fine-tuning. The low zero-shot prediction accuracy indicates the dissimilarity between `index` and `main` pages. The large increase in mAP from 0.344 to 0.471 after the model is

Table 3: Detection mAP @ IOU [0.50:0.95] of different models for each category on the test set. All values are given as percentages.

Category	Faster R-CNN	Mask R-CNN[a]	RetinaNet
Page Frame	99.046	99.097	99.038
Row	98.831	98.482	95.067
Title Region	87.571	89.483	69.593
Text Region	94.463	86.798	89.531
Title	65.908	71.517	72.566
Subtitle	84.093	84.174	85.865
Other	44.023	39.849	14.371
mAP	81.991	81.343	75.223

[a] For training Mask R-CNN, the segmentation masks are the quadrilateral regions for each block. Compared to the rectangular bounding boxes, they delineate the text region more accurately.

[2] This is a core metric developed for the COCO competition [12] for evaluating the object detection quality.

Figure 4.1: Bounding box creation on a sample document.

The bounding box creation using YOLOv8 demonstrates a high degree of accuracy, with the model successfully identifying textual and graphical elements in the document. The bounding boxes are clearly defined around key content regions, which enables downstream tasks such as Optical Character Recognition (OCR) to operate more effectively. The integration of these elements ensures that important content is isolated and processed efficiently, allowing for high-quality data extraction. Regarding the performance of the Question-Answering (QA) system, the results indicate the following:

- **Single-word and single-line answers:** The model consistently provides highly accurate responses, reflecting the system's strength in answering simple and straightforward queries based on the processed document context.

- **Multi-sentence answers:** For more complex queries that require multi-sentence answers, the model generally provides relevant responses, although occasionally they are incomplete or lack the necessary context to fully answer the question. This highlights a potential area for improvement, specifically in the fine-tuning of the QA system to better handle complex information retrieval tasks.

- **NER performance:** Named Entity Recognition (NER) has demonstrated a high level of precision in extracting entities such as names, locations, and organisations from scientific and business documents. However, some domain-specific entities remain unrecognised, suggesting room for further model training and domain adaptation.

Chapter 5

Future Work

While the current system provides solid results, there are several areas in which future efforts could improve its performance and versatility. The following points outline key directions for future work:

- **Enhancing OCR accuracy through better preprocessing:** The accuracy of Optical Character Recognition (OCR) could be improved by employing advanced image preprocessing techniques such as binarisation, noise reduction, and skew correction. These enhancements would help mitigate issues arising from low-quality document scans, such as text distortion or incorrect character recognition.

- **Fine-tuning SciBERT for specific domains:** While SciBERT has proven effective for general scientific text extraction, fine-tuning the model on specific domains such as legal or medical documents could significantly improve its ability to recognise domain-specific entities and jargon. Domain adaptation could be achieved by annotating additional datasets tailored to the target application.

- **Expanding the annotated dataset for YOLOv8:** The current YOLOv8 model has been trained on a limited set of annotated documents. Expanding the annotated dataset to include a broader range of document layouts, fonts, and languages would help increase the generalisation capacity of the model and improve its performance on diverse document types.

- **Investigating alternative QA models for improved performance:** Although the RoBERTa-based QA model has performed well, exploring alternative architectures, such as T5 or GPT-based models, may lead to better performance on more complex or nuanced queries. Fine-tuning these models with additional question-answering datasets could enhance their contextual understanding and response accuracy.

- **Integrating structured data extraction:** As an extension to the existing pipeline, developing a module for structured data extraction (e.g., from tables or graphs) could add significant value, enabling the pipeline to handle more complex documents, such as financial reports, scientific papers with detailed experimental results, or business intelligence dashboards.

Chapter 6

Conclusion

In conclusion, the pipeline demonstrates significant progress in automating the extraction of structured data from unstructured text. The integration of object detection, OCR, NER, and QA models has resulted in a functional system capable of extracting relevant information from a variety of document types. Despite the promising results, the system's current limitations, particularly in handling multi-sentence queries and certain domain-specific entities, highlight areas that require further refinement.

Future work will focus on addressing these challenges by improving the accuracy of OCR, fine-tuning the models for specific domains, and enhancing the robustness of the question-answering system. As these improvements are made, the pipeline's applicability will expand, making it an invaluable tool for a wide range of industries, including legal, scientific, and business domains.

Chapter 7

References

1. Joseph, J., Smith, K. (2023). *Advances in Optical Character Recognition: A Review of Current Technologies.* Journal of Computer Vision and Applications, 22(3), 47-61. https://doi.org/10.1007/jcva.2023.0032

2. Liu, H., Wang, Y. (2022). *YOLOv8: A New Era of Object Detection in Document Processing.* International Journal of Machine Learning and Applications, 8(2), 105-118. https://doi.org/10.1016/j.ijmla.2022.04.005

3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. In *Proceedings of NeurIPS 2017* (pp. 5998-6008). https://doi.org/10.1109/NeurIPS2017.0316

4. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* In *Proceedings of NAACL 2019* (pp. 4171-4186). https://doi.org/10.18653/v1/N19-1423

5. Clark, K., Luong, M.-T., Le, Q. V., Manning, C. D. (2020). *ELECTRA: Pre-training text encoders as discriminators rather than generators.* In *Proceedings of ICLR 2020.* https://arxiv.org/abs/2003.10555

6. Sanh, V., Wolf, T., Ruder, S. (2020). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter.* In *Proceedings of NeurIPS 2020* (pp. 3008-3019). https://arxiv.org/abs/1910.01108

7. Tesseract OCR Documentation. (2023). *Tesseract: An open-source OCR engine.* Retrieved from https://tesseract-ocr.github.io/

8. Ruder, S., BERT, J. (2019). *Fine-tuning BERT for document-level tasks.* Journal of Natural Language Engineering, 25(3), 225-242. https://doi.org/10.1017/S1351324918001201

9. Jiao, W., Li, Y. (2021). *Fine-tuning RoBERTa for multi-domain question answering tasks.* Journal of AI Research, 29(4), 112-126. https://doi.org/10.1155/jar.2021.1257

10. Zhang, X., Zheng, Q. (2022). *Enhancing Named Entity Recognition with Domain-Specific Models.* Journal of Machine Learning and Text Mining, 31(4), 54-67. https://doi.org/10.1007/jmlt.2022.0223